

ČERVENEC

FUN

1993

with Commodore

časopis uživatelů Commodore 64/128

7. číslo

BASIC

Tipy a triky

Programujeme DEMO

GEOS 1.3

SID - Symphony

Vyhodnocení
čtenářské
ankety

OBSAH

Úvodní slovo	1
Basic	2
Assembler na C64	5
Tipy a triky	7
Programujeme DEMO	9
Představujeme:	
GEOS 1.3	11
Přečetli jsme za Vás:	
SID – Symphony	17
Dopisy čtenářů	18
Comotronic News:	
Vyhodnocení čtenářské ankety	22
Inzerce	24

CO BUDE V PŘÍŠTÍM ČÍSLE?

Basic – pokračování

Assembler – pokračování

Tipy a triky

Programujeme DEMO (dokončení)

C64 a video

Představujeme:

ČStext

PLOT BASIC

Comotronic News

Dopisy čtenářů

Inzerce

Vážený čtenáři,

Commodore 64 se osvědčil jako všestranný pomocník všech zvědavých uživatelů, kteří se s ním blíže seznámili. Slouží neúnavně jako herní konzola (to asi nejčastěji), ale také jako hudební nástroj, slovník, inteligentní psací stroj, stroj na zpracování dat nejrůznější povahy. Slouží pro řízení, měření, regulaci a má svoje uplatnění i v řadě dalších aplikací. Jednou z oblastí, které se bez výpočetní techniky dnes již neobejdou, je také video.

Video zaznamenává v posledních létech prudký nárůst zájmu i u nás a dostává se pomalu ale jistě z oblasti pasivního nahrávání a přehrávání televizních pořadů a filmů do oblasti aktivní tvorby. Počet videokamer, které jsou u nás v užívání neustále roste a s nimi roste i počet těch, kteří mají zájem nejen něco natočit, ale vytvořit dílko kompletní, včetně titulků a vkládaných textů. Pro tyto zájemce jsou k dispozici různá triková a titulkovací zařízení, mixážní pulty atd.

Po čem každý vyspělejší videoamatér touží, je zařízení, které by mu umožnilo vkládat obraz do obrazu. Takové zařízení pracující ve spojení s počítačem, což mu dává možnost ještě nejrůznějších dalších triků se nazývá GENLOCK. Znájí jej majitelé Amigy a v poslední době o něm mohou uvažovat i majitelé C64, neboť GENLOCK pro C64 uvedla na trh firma SCANNTRONIK z NSR. Můžeme Vám prozradit, že elektronickou část Genlocků pro C64 vyrábí naše firma.

Ale vraťme se na pevnou zem, Genlock pro C64 stojí 850,- DM a proto u nás asi nebude příliš rozšířen. Existuje však i jednodušší i když ne tak mnohostranné řešení, kterým je program VIDEOFOX nebo modul VIDEOPROFI a další programy, které jsou podstatně levnější a i tak skýtají široké možnosti tvorby titulků a obrazů. Proto od příštího čísla zavádíme pravidelnou rubriku **C64 a video**, která se bude zabývat uvedenou problematikou od nejjednoduššího programu, až po to nejdokonalejší zařízení, které je pro C64 k dispozici. Věříme, že tím rozšíříme Vaše možnosti využití nestárnoucího C64.

Jiří Kouřil
redaktor 7. čísla

HRA O ČAS

Programy v BASICu se dají optimalizovat třemi různými směry.

Jednak tak, aby byly dobře čitelné, potom tak, aby byly rychlé a nakonec tak, aby šetřily místo v paměti. Že to není jednoduché, vyrovnat se přijatelně s uvedenými kritérii, zná každý, kdo strávil programováním alespoň pár hodin. Ukážeme si dále, jak se s těmito často protichůdnými požadavky rozumně vyrovnat.

Commodore 64 má zabudovaný systémové hodiny, které se dají z Basicu snadno dotazovat pomocí systémové proměnné TI\$. Pomocí těchto hodin budeme dále kontrolovat, jaký mají různé úpravy programů vliv na jejich rychlost. Abychom se dobrali stejných výsledků, je nutné, abyste dále použité programy zapisovali přesně tak, jak jsou uvedeny. Taky je nutno, abyste předem vypnuli nebo odpojili všechna hardwarová rozšíření jako Simon's Basic nebo FCIII, které mohou taky ovlivňovat výsledek.

Stopky

Před spuštěním programu musíme vždycky nastavit hodiny na 0 a na konec se dotázat, kolik času program potřeboval ke svému provedení. V rezervované proměnné TI se nachází čas v 1/60 sekundy. Pokud hodnotu této proměnné vydělíme 60, získáme čas v sekundách. To využijeme v našich kontrolních programech. Proto budou všechny obsahovat dále uvedené řádky 10 a 90.

```
10 TI$ = "000000"
90 PRINT TI/60 "SEKUND"
```

Mezi nimi pak budou uvedeny vlastní řádky programů. Pokud odstartujeme uvedené dva řádky příkazem RUN, objeví se nám výsledek:

0 SEKUND. To znamená že dobu, kterou potřebují k provedení řádky 10 a 90 můžeme zanedbat.

Při zkouškách jsme zjistili, že zlepšení, dosažená úpravou programů dosahují časů v desetinách nebo setinách sekundy. To se zdá být mizivě málo, ale uvědomte si, že tyto časy se sčítání a smyčky například FOR – TO – NEXT mívaly i tisíce průchodů v jednom programu. Pak je i zkrácení jednoho průběhu o setiny sekundy významné, neboť zkracuje běh celého programu o desítky sekund.

Neukážeme vám dále, jak promyšleným řešením postupu výpočtů nebo pomocí strojových rutin se dá program urychlit, ale například jak jsou dvě verze jednoho příkazu rozdílné z hlediska časové náročnosti.

Například příkaz

```
A = 25731*14
```

je zpracován rychleji než příkaz

```
A = 14*25731,
```

ačkoliv jsou matematicky identické, protože oba dají stejný výsledek, číslo 360234.

Protože pro měření rychlosti používáme systémové hodiny, které běží s rozlišením jen 1/60 sekundy, budeme muset průběh programů řádově tisíckrát opakovat za pomoci smyčky FOR – TO – NEXT. Už při zápisu této smyčky je však možno ušetřit nějaké ty tisíce sekundy.

Zkuste například program

```
20 FOR I=1 TO 5000: NEXT I
```

Nevkládejte do řádku žádné prázdné znaky! Po startu pomocí RUN naměří počítač přesně 6,65 sekund.

Nyní vymažte I za příkazem NEXT a spusťte program znovu. Je to těžko k uvěření, ale čas se zkrátil na 5.47 sekund. To je zisk na rychlosti 18% a to pouhým vynecháním nepotřebného znaku.

Pokud výše uvedenou smyčku rozepíšete na 2 řádky:

```
20 FOR I=1 TO 5000
```

```
30 NEXT I,
```

Průběh se prodlouží na 6.83 sekundy, to o 2,7% proti jednořádkové verzi.

TIP 1

Více příkazů v jednom řádku zpracuje počítač rychleji, než stejné příkazy, rozepsané do více řádků.

Taky v tomto případě lze vypuštěním proměnné I ušetřit čas. Doplňte program ještě poznámkou v řádku REM, například

```
25 REM ** TEST **
```

Čas se prodlouží na 9.53 sekundy, to je prodloužení o 40%! Divíte se teď, proč mnozí programátoři zcela zapomínají na řádky REM?

A pokud se domníváte, že příčina je jen v zavedení dalšího řádku, přepište řádek 20 na

```
20 FOR I=1 TO 5000: NEXT: REM ** TEST **
```

Zlepšení na 9.35 sekundy nestojí vlastně za řeč.

Teď si zkusíme jeden technický trik. Zapište program

```
15 POKE53265,0
20 FORI=1TO5000
35 POKE53265,27
```

Poznámku REM v řádku 20 jsme vypustili, svůj účel už splnila, ukázala nám, jak zdržuje program. Než upravený program spustíte, musíme objasnit pouky na řádcích 15 a 35.

V počítači C64 je zabudován obvod, který je zodpovědný za vytváření obrázku na obrazovce – videočip VIC. Vytváření obrazu není práce jednoduchá a VIC proto pro svou práci spotřebovává část systémového času. Pomocí pouky v řádku 15 proto VIC vypneme (nastavíme barvu obrazovky na barvu okolí). Při práci počítače ve smyčce, kterou program pokračuje, se na obrazovce stejně nic podstatného neděje, takže si to můžeme dovolit. A když práce ve smyčce skončí, můžeme VIC opět zapnout, k čemuž slouží řádek 35. Výsledek: jen 5.17 sekund! Pro srovnání – bez řádků 15 a 35 trvala smyčka 6.85 sekund. Ušetřili jsme dobrou čtvrtinu času.

TIP 2

V těch částech programu, kde probíhají časově náročné výpočty a obrazovka nemusí být vidět, se vyplatí vypínat VIC. To nemá žádný záporný vliv na program, které běží i s vypnutou obrazovkou dál.

Nyní se podíváme na proměnné, bez kterých se žádný program neobejde. Právě zde se dá ušetřit spousta času, když známe několik základních triků.

Vymažte nejdříve všechny řádky kromě 10 a 90. Pak zapište nové řádky 20 a 40:

```
20 FORI=1TO2000
40 NEXT
```

Budeme tedy 2000krát provádět nějakou operaci. Například nastavení proměnné A na hodnotu 0. Podle příručky se to dělá takto:

```
30 LET A=0
```

Když takto doplníme smyčku a spustíme program, dostaneme čas 5.82 sekund. Ale skoro žádný programátor nepoužívá slovíčko LET, píše tedy řádek 30 takto:

```
30 A=0
```

Takto upravená smyčka způsobí zkrácení běhu programu na 5.72 sekund. Počítač totiž nemusí vyhodnocovat význam znaků LET.

Těžko objasnitelné je, jak funguje následující geniální trik. Ale výsledek se opravdu nechá vidět:

```
30 A=.
```

Držte se pevně – jen 4.53 sekundy! Pouhé nahrazení nuly tečkou vedlo ke zkrácení času o 22 procent.

TIP 3

U dekadických čísel se vyplatí nahrazovat nulu tečkou.

Poke 53280,0 je pomalejší než POKE 53280.

Nula před desetinou tečkou také zpomaluje provádění programu A=0.553 je pomalejší než A=.553

Také další trik není všeobecně znám: definujte proměnné hned na začátku programu – významně to zkrátí běh programů. V našem případě například definujte proměnnou A:

```
5 A=0
```

Tím se na běhu programu nic nezmění, ale doba běhu se zkrátí na 4.46 sekundy. Obecně se vyplatí často používané proměnné v programu nejdříve inicializovat (tak se nazývá přiřazení hodnot proměnným na začátku programu).

Tato verze řádku 5 změní dobu běhu na 4.62 sekund:

```
5 B=0:C=0:A=0
```

Důvod je nasnadě: při přiřazování hodnoty 0 proměnné A v řádku 30 kontroluje počítač nejdříve ve své tabulce, zda už proměnná A existuje. V první verzi řádku 5 byla proměnná A na prvním místě, proto ji počítač našel rychleji, než v druhém případě, kdy se musí k proměnné A prokousávat přes zbytečné iniciace B a C, což nějaký čas trvá.

Názvy proměnných mohou být libovolně dlouhé, například P, ale I PROMENNA. Počítač C64 však „bere“ za označení maximálně 2 znaky, proto jsou ostatní zbytečné. Pro něj je PR a PROMENNA to stejné označení. Ale jinak je tomu z hlediska času.

Upravíme řádek 30 do podoby

```
30 PR=0
```

Program spustíme a dostaneme čas 5.85 sekundy.

Pak řádek změním na

```
30 PROMENNA=0
```

A vida! Počítač se s touto změnou opět vyrovnává jen za cenu prodloužení doby běhu programu. Čas se prodloužil na 6.6 sekund, což není jistě zanedbatelné. Samozřejmě nejrychlejší je program, když jsou použita jen jednoznaková označení proměnných.

```
30 P=0
```

Výsledek je jasný – 5.73 sekundy mluví za vše. Proto používejte v programech co nejkratší označení proměnných.

A nyní ještě jeden trik, kterému je těžko uvěřit. Ale zkuste si to, když nevěříte!

```
5 A=0
```

```
20 FORI=1TO1000
```

```
30 A=14*25731
```

```
40 NEXT
```


Tento nesmyslný program přiřazuje proměnné A hodnotu násobení $14 \cdot 25731 = 360234$. Programu to trvá 11.67 sekundy. Co budete dělat, když vám někdo bude tvrdit, že to musí jít rychleji? Ze zoufalství zkusíte třeba prohodit hodnoty 14 a 25731:

```
30 A=25731*14
```

Matematicky se na zadání nic nezměnilo, ale čas ano: 11.33 sekundy!

Nyní se dostáváme k triku, který způsobuje velkou úsporu času. Jde o postupy umocňování. Až odstartujete program s upraveným řádkem

```
30 A=7^2
```

příkazem RUN, najděte si dobrou knížku, přečtěte si dalších pár stránek z tohoto časopisu nebo si uvařte kávu. Budete na to mít dost času. Počítači to potrvá skoro minutu, než program ukončí.

Ale druhá mocnina sedmi se dá spočítat také jinak:

```
30 A=7*7
```

Nechce se věřit, ale teď to trvalo opravdu jen 5.17 sekundy!

TIP 4

Vyhnete se v programu umocňování, jak jen to bude možné. Nejenom proto, že trvá desetkrát tak dlouho, jako násobení, ale také proto, že je nepřesné.

Zadejte ještě jednou

```
PRINT 7^2
```

a potom

```
PRINT 7*7
```

Výsledek není nutno komentovat...

Nyní si pohrajeme s konstrukcí IF...THEN. Vymažte opět všechny řádky kromě 10 a 90, potom zapíšte tyto:

```
5 A=0:B=0
```

```
20 FOR I=1TO50
```

```
30 IFA=0ANDB=0THEN40
```

```
40 NEXT
```

Ten příkaz v řádku 30 je nesmyslný, pokud je A i B rovno 0, pak se má skočit na řádek 40, což však nastane vždy. Ale jde jen o příklad. Počítač se s programem vyrovná za 9.2 sekundy. Nyní se postaráme o to, že jedna z podmínek nebude splněna:

```
5 A=1:B=0
```

Protože nyní se neprovádí příkaz za podmínkou THEN, zkrátí se doba běhu programu: 7.8 sekundy. Řádek 30 zkoumá pokaždé, zda jsou splněny obě podmínky, což je zbytečné. Pokud je A různé od 0, může se řádek 30 rovnou přerušit. K tomu slouží následující úprava:

```
30 IFA=0THENIFB=0THEN40
```

Takto upravený program běží již jen 3.98 sekundy, tedy přibližně poloviční dobu. Tato verze se projeví kladně i v první variantě, kdy je řádek 5 upraven do podoby

```
5 A=0:B=0
```

Doba běhu je 7.12 sekundy, což je proti původním 9.2 sekundám úspora 20% proti původní verzi řádku 30.

Pravidla pro rychlé programy

Dále uvedené zásady pomáhají urychlit programy v BASICu. Dbejte na ně zvláště ve smyčkách, kde mají největší efekt.

1. Často používané proměnné a konstanty se definují hned na začátku programu.
2. Názvy proměnných volíme pokud možno jednomístné, nejvýše dvoumístné.
3. Při testech IF...THEN s více podmínkami používejte více příkazů IF za sebou. Podmínky řaďte za sebou od nejméně pravděpodobné po nejpravděpodobnější, pokud je to možné.
4. Pokud je to možné, dávejte přednost smyčkám FOR...NEXT před smyčkami IF...THEN, jsou rychlejší.
5. V příkazu NEXT vynechejte název proměnné, pokud je to možné.
6. Programy bez REM řádku a poznámek a bez mezer mezi příkazy jsou rychlejší. Takže především ve smyčkách se REM vyhnete!
7. Pro omezení počtu řádku zapisujte do jednoho řádku větší počet příkazů (oddělených dvojtečkou).
8. Při násobení zařazujte delší číslo před kratší.
9. Umocňování nahraďte vícenásobným násobením.
10. Místo samostatné nuly "0" zapisujte tečku ".".
11. Vynechejte nulu před desetinnou tečkou.
12. U časově náročných výpočtů vypínejte obrazovku.

(JK)

ASSEMBLER NA C64

(6. pokračování)

Posun bitů vlevo příkazem ASL

Před posunem:

Carry = 0, bit:	7	6	5	4	3	2	1	0
hodnota:	1	0	1	0	1	0	1	0

Po posunu:

Carry = 1, bit:	7	6	5	4	3	2	1	0
hodnota:	0	1	0	1	0	1	0	0

Posun vpravo pomocí příkazu LSR funguje analogicky. Bit 0 se přesune do příznaku Carry a na místo 7. bitu se zapíše 0.

Jak vyplývá z právě uvedeného, máme pro posun vlevo příkaz ASL (Arithmetic Shift Left) a pro posun vpravo LSR (Logical Shift Right).

Druhý typ příkazů pro posun jsou příkazy, které bity rotují v kruhu. V tomto případě není uvolněný bit vyplněn nulou, ale je do něj přenesen bit z příznaku pro přenos. Analogicky se pak „vystrčený“ bit přenesení do Carry.

U tohoto typu posunu je důležitá posloupnost jednotlivých procesů. Nejdříve se bajt posune a „vystrčený“ bit se interně uloží. Pak se obsah příznaku přenosu přesune do uvolněného bitu. Teprve pak se „vystrčený“ bit uloží do příznaku přenosu. Není tedy možné posunout bajt vlevo tak, že se 7. bit přenesení na místo uvolněného bitu 0. Jak se taková rotace provádí, bude probráno později.

Podívejme se nejdříve na schema posunu vpravo.

Posun vpravo příkazem ROR

Před posunem:

Carry = 0, bit:	7	6	5	4	3	2	1	0
hodnota:	1	0	1	0	1	0	1	0

Po posunu:

Carry = 0, bit:	7	6	5	4	3	2	1	0
hodnota:	0	1	0	1	0	1	0	1

Po posunu je v příznaku Carry uložen bit č.0 a obsah příznaku je přenesen na místo bitu 7. Posun vlevo funguje analogicky.

Pro posun vlevo máme v tomto případě příkaz ROL (ROtate Left) a pro posun vpravo příkaz ROR (ROtate Right).

Příklady použití:

Příkazy pro posun se často používají k zdvojnásobení nebo čtyřnásobení hodnoty a pod.

Pokud bajt posuneme o jedno místo vlevo, odpovídá to násobení 2. Analogicky tomu pak posun o další místo vlevo odpovídá násobení čtyřmi. Abychom tedy obsah akumulátoru znásobili čtyřmi, musíme jej posunout o 2 místa vlevo ($2 \times 2 = 4$) pomocí dvou příkazů ASL A (A za příkazem označuje, že se adresuje akumulátor). Možné přetečení musí být samozřejmě ošetřeno a přesunuto do dalšího bajtu, takže po ukončení posunu máme 16ti bitovou hodnotu. K tomu příklad:

Paměťové místo \$D8 obsahuje číslo \$FF, následující paměťové místo \$D9 obsahuje číslo \$00. Obsah paměťového místa \$D9 nyní znásobíme osmi ($2 \times 2 \times 2 = 8$). To provedeme následovně:

```
ASL $D8
ROL $D9
ASL $D8
ROL $D9
ASL $D8
ROL $D9
```

Jak tento program funguje?

Nejprve se obsah paměťového místa \$D8 posune o jedno místo vlevo. Na místo bitu 0 se uloží hodnota 0 a bit 7 se přesune do příznaku přenosu Carry. Aby bylo přetečení správně zpracováno, je nutné, aby se tento bit v následujícím příkazu posunul do dalšího bajtu. Tento úkol zajistí příkaz ROL \$D9. Obsah paměťového místa \$D9 se přitom posune o 1 místo vlevo a obsah příznaku přenosu se přenesení na pozici bitu 0. Když tuto sekvenci ASL/ROL provedeme třikrát za sebou, tak jsme obsah paměťového místa \$D8 znásobili osmi. Nicméně se jedná o šestnáctibitový výsledek, který je

umístěn ve dvou paměťových buňkách. \$D8 a \$D9 v pořadí LB-HB.

Jak již bylo v příkladu naznačeno, nabízí se v případě potřeby třeba cyklického posunu obsahu akumulátoru následující řešení:

LDA #%10101010 ;naplnění akumulátoru číslem, negativní příznak obsahuje hodnotu 7.bitu.
CLC ;vymazání příznaku Carry
BPL LABEL ;7.bit=0, pak skok

SEC ;pokud 7.bit =1, pak také Carry=1

LABEL ROL A ;rotovat akumulátor
Při rotaci se obsah příznaku Carry přesune do 0. bitu. Příznak ale obsahuje obsah negativního příznaku, který zase obsahuje obsah 7. bitu. Pokud se operuje s paměťovými místy, dojde uplatnění příkaz BIT (viz jeho popis v příslušné kapitole).

Nakonec ještě tabulka druhů adresování a příkazových kódů.

Druh adresování	příklad		ASL	LSR	ROL	ROR
akumulátoru	příkaz A		\$0A	\$4A	\$2A	\$6A
přímé	mnemokód	\$bajt	—	—	—	—
absolutní	mnemokód	\$adresa	\$0E	\$4E	\$2E	\$6E
absolutní indexované X	mnemokód	\$adresa,X	\$1E	\$5E	\$2E	\$7E
absolutní indexované Y	mnemokód	\$adresa,Y	—	—	—	—
nulté stránky	mnemokód	\$1b.adresa	\$06	\$46	\$26	\$66
nulté str.indexované X	mnemokód	\$1b.adresa,X	\$16	\$56	\$36	\$76
nulté str.indexované Y	mnemokód	\$1b.adresa,Y	—	—	—	—
nepřímé indexované	mnemokód	(\$1b.adresa),Y	—	—	—	—
indexované nepřímé	mnemokód	(\$1b.adresa,X)	—	—	—	—

Druh adresování „akumulátoru“ v tomto případě znamená, že příkaz pro posun se týká přímo akumulátoru. Tento druh adresování je v pravidlech rozlišen zápisem „A“ za slovo příkazu.

Příkazy pro STACK PHA, PLA, PHP, PLP, TXS, TSX

Abychom mohli řádně vysvětlit funkci příkazů pro STACK, musíme nejdříve vysvětlit, co to STACK vlastně je.

Pod STACKem rozumíme určitý paměťový rozsah paměti RAM, který je rezervován pro procesor. Tento rezervovaný paměťový rozsah začíná na adrese \$0100 a končí na adrese \$01FF. Zabírá tedy celou první stránku paměti. STACK má také význam hromady, v našem případě hromady číselných hodnot.

Procesor užívá STACK zejména pro ukládání návratových adres při volání podprogramů a k přechodnému uložení obsahu registrů. Aby bylo možno zjistit, na které místo má být uložena další hodnota, existuje takzvaný ukazatel STACKu, STACK POINTER (SP). Ten ukazuje vždy na další volnou adresu ve STACKu. Pokud do STACKu uložíte pomocí příslušného příkazu nějakou hodnotu, uloží se tato hodnota na adresu \$0100 + obsah ukazatele STACKu. Když k tomu dojde, je zároveň hodnota ukazatele STACKu zmenšena o 1,

takže ukazuje opět na další volnou pozici STACKu. Pokud je ze STACKu hodnota přesouvána do registru, dochází k opačnému ději, t.zn., že se hodnota ukazatele zvýší o 1.

Při práci se STACKem je nutno pamatovat na skutečnost, že pracuje systémem LIFO:(LAST IN – FIRST OUT), to znamená, že při ukládání více hodnot do STACKu a jejich následujícím odebírání se jako první odebere hodnota, která byla uložena jako poslední. Tento princip také osvětluje označení STACK – hromada. Pokud ukládáme věci na hromadu, je poslední uložená věc nahoře. A pak při odebírání věcí z hromady se také nejdříve odebírá to co je nahoře a potom to další. Pro větší názornost si uvedeme ještě příklad, kdy do STACKu uložíme 3 hodnoty a pak je opět odebereme (přemístíme do registru).

Ukazatel STACKu (SP) nechť je na pozici \$03 a naše 3 hodnoty jsou \$10, \$A0 a \$DC.

Před tím, než do STACKu začneme ukládat naše hodnoty, vypadá takto:

Pozice: \$0100 \$0101 \$0102 \$0103 \$0104 atd...
Ukazatel SP \$03

Nyní uložíme do STACKu první hodnotu. Ta se uloží do paměťové buňky, na kterou ukazuje SP, v našem pří-

padě na adresu \$0100 + SP = \$0103. Potom se obsah SP sníží o 1. Po této operaci vypadá takto:

```
Pozice: $0100 $0101 $0102 $0103 $0104 atd...
                $10
Ukazatel SP      $02
```

Uložení našich dalších dvou hodnot proběhne analogicky, takže po celé operaci bude námi sledovaný obsah STACKu vypadat takto:

```
Pozice:   $0100 $0101 $0102 $0103 $0104 atd...
                $DC  $A0  $10
Ukazatel SP $00
```

Pokud nyní budeme hodnoty ze STACKu číst a například ukládat do registru, bude jako první přesunuta hodnota \$DC, hodnota SP se změní na \$01. Odečítání dalších hodnot ze STACKu bude probíhat obdobně. Jako druhá se ze STACKu přesune hodnota \$A0 a jako třetí \$10.

Uvedený příklad názorně ukazuje, jak systém LIFO pracuje.

Nyní, když víme, jak STACK pracuje, můžeme si vysvětlit funkci a použití příkazů pro práci s ním.

(JK)
pokračování

TIPY A TRIKY

1. ABSOLUTNÍ LOAD BEZ MONITORU

Většina uživatelů C64 bere jako fakt, že natáhnout data do počítače od přesně stanovené adresy je možné jenom v Monitoru. Ale s pomocí dále uvedeného programku to jde i z BASICu! Dále uvedené parametry zLO a zHI jsou LOW a HIGH bajty počáteční adresy pro ukládání.

```
10 SYS 57812 "NÁZEV",8: POKE 780,0
20 POKE 781, zLO: POKE 782, zHI
30 SYS 57717: REM LOAD-START
```

2. ABSOLUTNÍ SAVE BEZ MONITORU

Je jasné, že pokud je možné bez Monitoru natahovat, půjde také ukládat. Stejně jako v předchozím případě jsou důležité adresy, odkud a pokud se má obsah paměti uložit. K tomu účelu jsou rozděleny počáteční a koncová adresa na části zLO/zHI a kLO/kHI. A nyní potřebný program.

```
10 SYS 57812,"NÁZEV",8
20 POKE 174, kLO: POKE 175, kHI
30 POKE 193, zLO, POKE 194, zHI
40 SYS 62957: REM SAVE-START
```

Příklad: potřebujeme uložit obsah paměti od adresy 20000 po 22000 na kazetu pod názvem "SPRAJTY"

```
10 SYS 57812 "SPRAJTY",1
20 POKE 193,32: POKE 194,78
30 POKE 174,241:POKE 175,85
40 SYS 62957
```

Z příkladu je jasné, že adresa se musí rozdělit na dvě hexadecimální části a ty přepočíst na dekadické, které se pak ukládají do paměti.

Jako koncovou adresu musíme jako obvykle při ukládání z monitoru uvést adresu+1.

3. 16 FUNKČNÍCH KLÁVES

Pomocí geniálního triku je možno i u C64 z programu dotazovat 16 funkčních kláves! A navíc se obejdeme bez obvyklého dotazu IF-THEN-ELSE.

Nejdříve si v programu definujeme funkci:

```
10 DEFFNA(X)=(X>2)*(X<7)*(((X-3-(X<4)*4)*2)
+(Y=0 OR Y=2)-(Y>1)*8)
```

Na odpovídajícím místě programu pak zařadíme dotazovací řádek:

```
100 X=PEEK(197):Y=PEEK(653):IFFNA(X)=0
THEN 100
```

110 pokračování v programu

Pokud bude stisknuta některá z funkčních kláves, bude program pokračovat dále na řádce 110. V proměnné X budete nyní mít až 16 různých hodnot, podle toho, které kombinace kláves stisknete. K dispozici jsou tyto funkční klávesy:

F1, F3, F5, F7 samotné,
F1, F3, F5, F7 v kombinaci se SHIFT,
F1, F3, F5, F7 v kombinaci s CBM,
F1, F3, F5, F7 v kombinaci s CTRL.

4. TÓN ZVONKU

Pokročilejší programátoři uživatelských programů se většinou málo zajímají o svět zvuků. Proto v jejich programech většinou chybí akustická signalizace stisku kláves. Proto dále uvádíme hotovou rutinu, která imituje tón zvonku a je vhodná k zařazení do vašich programů.

10 S=54272

20 POKE S+24,15: POKES+1,110: POKE S+5,9

30 POKE S+6,9: POKE S+4,17: POKE S+4,16

5. JEDNO NEBO DRUHÉ, ALE NE OBĚ.

Dobří programátoři v Basicu dokáží svoje programy zkrátit využitím logických operátorů AND a OR. Je však nutno přiznat, že C64 chybí ještě jeden důležitý logický operátor – EOR (exclusive or).

Při AND je výsledek 1, když oba operandy jsou rovny 1.

Při OR je výsledek 1, když jeden z operátorů je 1.

Co chybí, je logické spojení EOR, které dává výsledek 1, když přesně jeden z operandů je roven 1, ale ne oba. EOR je však možno simulovat s pomocí operátorů, které má C64 k dispozici tímto způsobem:

$X = (A \text{ OR } B) - (A \text{ AND } B)$

X je potom výsledek logické vazby EOR mezi A a B.

6. VÍCE PAMĚTI PRO BASIC

Programátoři v Basicu, kteří pracují s mnoha proměnnými se často dostávají do konfliktu s omezenou kapacitou paměti počítače. Těm pak doporučujeme využít paměť od adresy 49152. Malý příklad pro názornost uvádíme dále. Listing 1 ukazuje klasické řešení problému, listing 2 pak řešení, nezatěžující RAM pro Basic další proměnnou, ale využívající volnou paměť počítače.

0 REM LISTING 1

10 A=5

20 B=A+2

30 IF A>7 THEN PRINT A

100 REM LISTING 2

110 POKE 49152,5:

120 POKE 49153,(PEEK(49152)+2)

130 IF PEEK(49152)>7 THEN PRINT PEEK(49152)

7. VZDÁLENOST JE DŮLEŽITÁ

Disketová jednotka 1541 nesnáší blízkost monitoru nebo televizoru. Často se nedají programy vůbec natahnout a jindy se při natahování stále hlásí chyby.

Přitom stačí posunout diskovku kousek dál od monitoru a vše je zase v pořádku. Dodržujte proto "bezpečnostní" vzdálenost mezi monitorem a diskovkou alespoň 30 cm.

8. PŘEPÍNÁNÍ ADRES U DISKETOVÉ JEDNOTKY

Na sériovém kanálu počítače C64 se dají provozovat maximálně 4 diskovky najednou. Každá z nich ale musí mít přiřazenu jinou adresu. K tomu jsou nové diskovky VC1541-II vybaveny vzadu dvoupáčkovým přepínačem DIP. Adresy 8 – 11 se volí následující polohou přepínačů:

přepínač 1	přepínač 2	adresa (periferie)
nahoru	nahoru	8
dolů	nahoru	9
nahoru	dolů	10
dolů	dolů	11

Pokud nastavíte jinou než základní polohu přepínačů, bude diskovka reagovat jen v případě, že při komunikaci počítače s ní uvedete správné číslo periferie (adresu).

9. AUTOSTART JEDNODUŠE

Kdo je líný psát příkazy na obrazovku, například po nahrání hry zapisovat příkaz RUN a mačkat pak RETURN, může využít kombinace kláves SHIFT + RUN/STOP. U datassetu způsobí tato kombinace natažení programu a autostart, u diskovky je nutno napsat obvyklý příkaz LOAD"název",8 a stisknout uvedenou kombinaci. Autostrat je zajištěn.

10. LETADLO

Nakonec pro vás máme něco zvláštního. Zapište do počítače dále uvedený program a spusťte jej příkazem RUN. Z reproduktoru se ozve hluk letadla a ačkoliv je C64 již zpátky v přímém módu, zvuk se stále mění. Proč asi?

0 REM LETADLO

1 DATA 1,4,0,2,65,240,251,0,4,0,2,65,240,251,32

2 DATA 89,166,76,174,167,0,0,20,243,31

3 FOR I=. TO 24: POKE 54272+I,..:READ X: POKE 54272+I: NEXT

(JK)

PROGRAMUJEME DEMO

Chtěli byste také programovat DEMO na vašem C64? Já vím, na první pohled vypadají DEMO programy tak nepochopitelné, že si skoro každý řekne, že to nikdy nedokáže. Ale ruku na srdce, ty DEMO programy, které vypadají tak složitě, také programovali jen lidé a ne kouzelníci. Proto se domnívám, že stojíte za to, seznámit Vás se základy programování takových programů. Kdo ví, možná že pak opravdu ztratíte respekt před dly ostatních a pustíte se do programování sami.

Na začátku každého DEMO programu stojí nápad. Ať je revoluční nebo obdobný tomu, co zde již bylo, je v zásadě jedno. V každém případě je před vámi spousta programátorské práce. A ještě jednu radu. Nechtějte toho zvládnout příliš mnoho najednou. Postupujte pomalu od jednodušších věcí po složitější. Uvidíte, že se to vyplatí.

Z těchto částí se budeme nejdříve zabývat přerušitelným IRQ, které je pro DEMO programy nejdůležitější a má i mnoho dalších možností využití.

IRQ RUTINA

Pokud má rutina IRQ zajišťovat více než jednu funkci, bude pro začátečníka hodně tvrdým oříškem. Aby vám celá záležitost nepřipadala tak složitá, nabízíme vám příklad, na kterém je funkce zpracování přerušování IRQ vysvětlena. Ovladač IRQ může volat až 8 různých rutin po sobě (v té samé smyčce IRQ), přičemž je deaktivace možná nejjednodušeji uvnitř jedné (sub)rutiny. Nejdůležitější proměnná je MAINMASK (adresa 124). Je vybudována jako bitově orientovaná, to znamená, pokud je bit 0 nastaven, provede se rutina 1, pokud je bit 1 nastaven, provede se rutina 2, jinak ne, atd.

bit	7	6	5	4	3	2	1	0
rutina	8	7	6	5	4	3	2	1

Nejvyšší prioritu má bit 0, potom 1 a nakonec bit 7. Aby požadovaná rutina proběhla, musí být odpovídající bit nastaven. K tomu musí být zároveň nastavena proměnná MAINCOUNT na 0. Nahoře dotazované rutiny musí být subroutiney, to znamená, že musí končit příkazem RTS. Pokud napíšete rutinu která má být zabudována do IRQ, musí se jen přenést adresa do ROUTTAB a nastavit odpovídající bit v MAINMASK. Pro adresy v ROUTTAB platí:

rutina 1 → ROUTTAB;
rutina 2 → R1; atd.

Nakonec musíme ještě vektor IRQ nasměrovat na MAINIRQ a požadované úkoly budou v IRQ provedeny.

PŘÍKLAD – Listing ovladače IRQ jako zdrojový kód (Hypra-Ass-formát)

```

10 -      .ba $c000          ;startovní adresa
15 -                                ;49152
20 -      .gl irqvec=$0314
30 -      .gl irqold=$ea31
40 -      .gl mainmask=124
50 -      .gl maincount=125
60 -      .gl mainsave=126
70 -
80 - setirqnew sei          ;zabránění přerušování
90 -      lda #<(mainirq)   ;lo - a
100-      ldx #<(mainirq)   ;hi bajt
105-                                ;vektoru IRQ
110-      sta irqvec        ;nasměrování
120-      stx irqvec+1      ;na ovladač
125-                                ;IRQ
130-      ldy #00           ;maincount
140-      sty maincount     ;nastavit na 0
150-      sty 198           ;žádná klávesa
155-                                ;v kláv.bafru
160-      lda #%00000111    ;mainmask pro
170-      sta mainmask      ;nastavení 3. rutin
180-      ldx #64           ;žádná stisknutá
190-      stx 203           ;klávesa
200-      cli               ;uvolnit znovu IRQ
210-      jmp irqold        ;volání staré
215-                                ;rutiny IRQ
220-
230- racolor inc 53280       ;zvýšení barvy
235-                                ;rámečku
240-      rts               ;zpět
250-                                ;
260- bcolor inc 53281        ;zvýšení barvy
265-                                ;pozadí
270-      rts               ;a zpět
280-      ;
290- klávesu lda 203         ;klávesa f1
300-      cmp #4            ;stisknuta?

```


310-	beq tl	;když ano, potom tl	710-r1	.va bcolor	;rutina pro
320-	rts	;jinak zpět	715-		;nastavený bit 1
330-tl	sel	;zabránění přerušení	720-r2	.va key	;rutina pro
340-	lda #00	;mainmask a	725-		;nastavený bit 2
350-	sta mainmask	;mainsave nastavit	730-r3	.va 0	;bit 3
360-	sta mainsave	;na 0	740-r4	.va 0	;bit 4
370-	cli	;připustit irq	750-r5	.va 0	;bit 5
380-	rts	;zpět	760-r6	.va 0	;bit 6
390-		;	770-r7	.va 0	;bit 7
400-mainirq	lda mainmask	;masku natáhnout,			
410-	beq endirq	;když 0 endirq			
420-ml	lsl	;bit 0 nastaven?			
430-	bcs m2	;když ano m2			
440-	inc maincount	;zvýšit maincount			
450-	inc maincount	;o 2			
460-	ldx maincount	;maincount do x-reg.			
470-	cpx #16	;>=16			
480-	bcs endirq	;ano, ednirq			
490-	jmp ml	;ne, ml			
500-		;			
510-m2	ldx maincount	;maincount do X-reg.			
520-	sta mainsave	;aktuální mainmask			
525-		;uložit do mainsave			
530-	lda routtab,x	;hl-byte rutiny			
535-		;vytáhnout			
540-	sta irqout+1	;a přímo v kódu			
545-		;uložit			
550-	inx	;zvýšit x-reg.			
560-	lda routtab,x	;lo-bajt rutiny			
565-		;vytáhnout			
570-	sta irqout+2	;a opět přímo v kódu			
575-		;zapsat			
580-	jmp irqout	;skok na irqout			
590-		;			
600-endirq	lda #00	;maincount nastavit			
610-	sta maincount	;0 znovu nastavit			
620-	jmp irqoid	;volání staré			
625-		;rutiny IRQ			
630-		;			
640-irqout	jsr \$ffff	;volání rutiny			
645-		;jako subrutiny			
650-	lda mainsave	;opět natáhnout			
655-		;předchozí mainmask			
660-	inc maincount	;zvýšit čítač			
670-	inc maincount	;o 2			
680-	jmp ml	;zpět na ml			
690-		;			
700-routtab	.va racolor	;rutina pro			
705-	;nastavený bit 0				

NA ZÁVĚR JEŠTĚ REKAPITULACI

V řádcích 10 až 60 jsou uvedeny startovní adresy a proměnné. V řádku 80 je zamezeno přerušení a v řádcích 90 – 120 je odklon na ovladač IRQ. V řádcích 130 a 140 bude nastaven čítač maincount na 0. V řádcích 169 a 179 je nastavena MAINMASK. Jak se dá snadno poznat, jsou do IRQ zabudovány 3 rutiny (nastaveny bity 0 – 2). Adresy jsou uloženy v tabulce ROUTTAB od řádku 700. Rutiny RACOLOR a BKCOLOR zvyšují hodnotu barvy rámečku a pozadí, zatím co rutina KEY čeká na stisk F1. Pokud bude F1 stisknuto, deaktivují se všechny 3 rutiny, zabudované do přerušení. Nakonec je IRQ opět uvolněno (řádek 200) a pokračuje se dále se starou rutinou přerušení (řádek 210). Řádky 150, 180, 190 nastavují zpět všechna zadání z klávesnice.

A ještě pár poznámek pro Internisty

K funkci: nejdříve je MAINMASK přezkoušena na 0. Pokud nemá být v přerušení zpracována žádná rutina (MAINMASK = 0), skáče program přímo na ENDIRQ, kde se zpracovává normální rutina přerušení. Pokud je MAINMASK různá od 0, kontroluje ovladač IRQ, zda je nastaven první bit.

– pokud ano, uloží se hodnota čítače MAINCOUNT do registru X, akumulátor (změněná forma MAINMASK) se dočasně uloží do MAINSAVE, natáhne se adresa absolutně X-indexované rutiny, zapíše se za příkaz JSR u IRQOUT (samosemodifikující kód, protože není použit příkaz JSR (adresa) a nakonec skočí na JSR. Po zpracování vlastní rutiny uložíme do akumulátoru oper MAINSAVE, zvýšíme čítač a skočíme zpět na další zpracování.

– pokud ne, zvýší se čítač MAINSAVE o 2 a ukazuje, pokud je následující bit nastaven, na další adresu (LDA ROUTTAB,X). Pokud je hodnota čítače po zvýšení rovna 16, přezkouší se celý bajt a program skočí na ENDIRQ. Tam se čítač nastaví zpět a naskočí se k normální rutině IRQ.

(JK)

pokračování

GEOS V1.3

Na základě ohlasů k seriálu vycházejícím v minulém roce vznikl tento příspěvek. Pojednává o základech práce se systémem Geos 1.3. Program sám je dodáván na dvou disketách.

Disketa A

Strana 1: System Disk
systémové datové soubory (Geos Boot, Kernal, desk Top...)
budiče tiskáren
budiče myši, joysticku...
managery (foto, text, nastavení šipky, nastavení času...)
pomocné programy (budík, kalkulačka, poznámkový blok)

Strana 2: Disk Drivers
budiče tiskáren

Disketa B

Strana 1: Backup Sytem Disk
systémové datové soubory (Geos Boot, Kernal, desk Top...)
budiče tiskáren
budiče myši, joysticku...
managery (foto, text, nastavení šipky, na-

stavení času...)
pomocné programy (budík, kalkulačka, poznámkový blok)

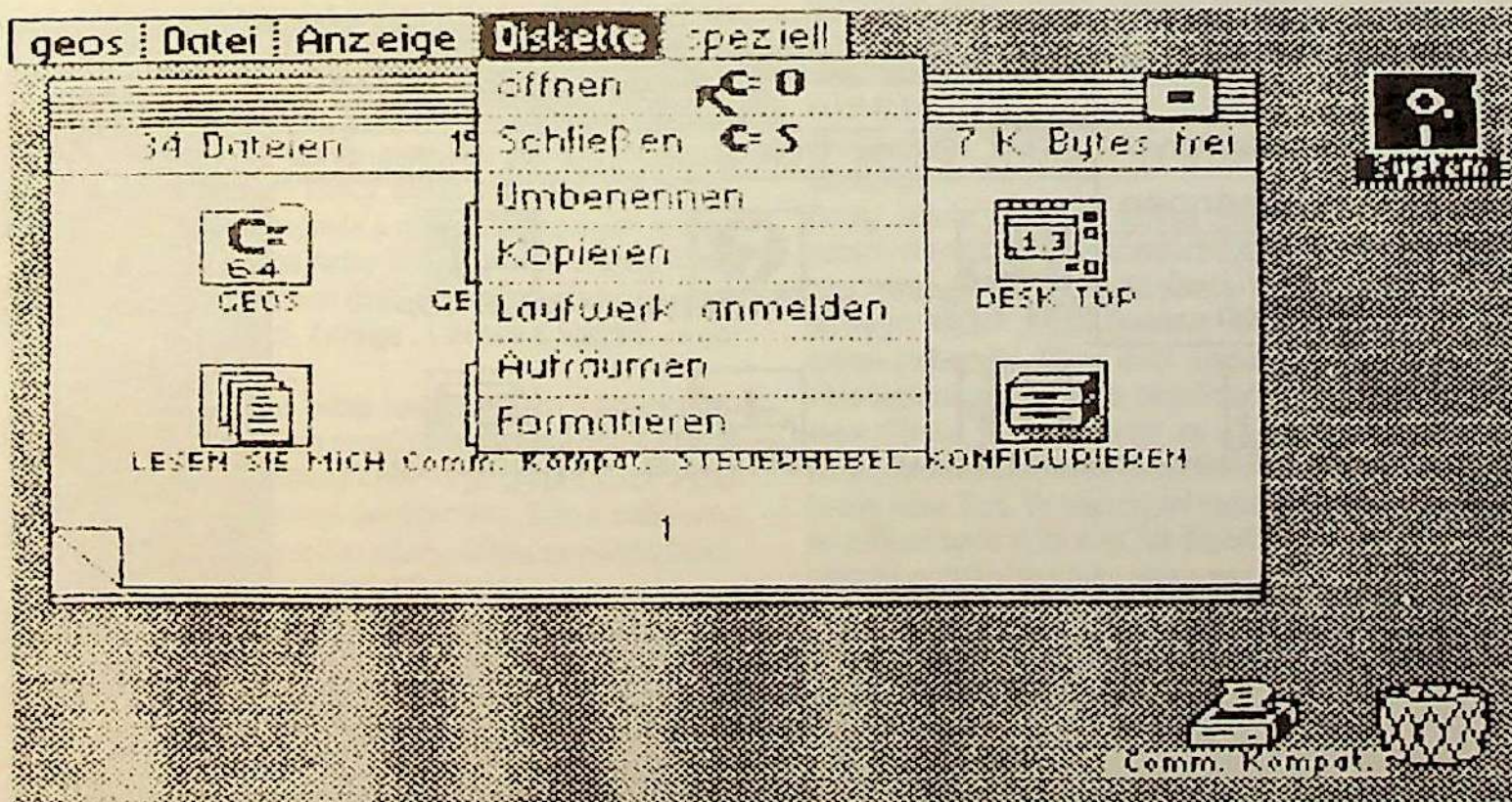
Strana 2: Applications Disk
desk Top
geoWrite V1.3
geoPaint V1.2
fonts (druhy písma)
pomocné programy (budík, kalkulačka...)
kopírovací program pro diskety

Na začátku práce práce s Geosem je nutno jej nainstalovat. Vezměte disketu označenou jako System Disk a vložte ji do dráivu. Klávesnicí počítače zadejte LOAD „GEOS“,8,1 a potvrďte stiskem Return. Dráiv začne pracovat a po chvíli se v komunikačním okně objeví výzva

**Bitte einlegen
Sicherungs-Systemdiskette.**

Vyjměte z přístroje disketu označenou jako System a založte místo ní Backup Sytem Disk. Pro potvrzení opět stiskněte Return. Po krátké práci disketové mechaniky se v komunikačním okénku objeví hlášení

**Bitte einlegen
Systemdiskette**



Vyměňte tedy opět Backup System Disk za System Disk. Po stisku Return se disketová mechanika dá do práce. Po krátkém probliknutí nápisu Geos bootet...

se po několika minutách objeví Geos deskTop. Modrou šipkou v levém horním rohu obrazovky přejeďte pomocí joysticku do pole Diskette a stiskněte tlačítko Fire. Rozbalí se roleta s řádkou funkcí. Vyberte Schliessen a opět stiskněte Fire. Piktogramy se symboly Geos, Kernal atd. zmizí. Na obrazovce zůstane prázdná plachta. Symbol diskety v pravém horním rohu, původně černý, nesoucí název System nyní zbělá a objeví se v něm otazník. Nyní je systém nainstalován a umožňuje další práci.

Ládování Geosu

Karetní hru s výměnou disket nebudete již nadále provádět. Po instalaci, která se mimochodem provádí pouze jedenkrát, se natahování systému do počítače zjednoduší na pár příkazů. Do dražvu vložte systémovou disketu a z klávesnice zadejte

LOAD „GEOS“,8,1

a stiskněte Return. Dražv se aktivuje. Na obrazovce se objeví hlášení

Geos bootet...

a po chvíli se ocitnete v úvodní obrazovce deskTop. Známým postupem, najet na políčko Diskette, zvolit Schliessen, uzavřete systémovou disketu a vyměňte ji za pracovní.

Upozornění! System Disk používejte pouze pro loading Geos do počítače. Dále pracujte výhradně s pracovními kopiemi disket.

PŘÍPRAVA PRACOVNÍ DISKETY

Před začátkem práce s uživatelskými programy běžícími pod Geos si nachystejte jednu nebo více pracovních disket. Postup jejich přípravy bude objasněn dále. Po naládování Geos a uzavření systémové diskety vložte do dražvu disketu Applications Disk. Disketu otevřete tím způsobem, že modrou šipkou najedete do políčka Diskette a stisknete knoflík joysticku. Vyberete políčko Offnen a opět stisknete knoflík. Na obrazovce se objeví deskTop diskety vložené do dražvu. Najedte šipkou na piktogram označený jako Diskettenkopier a jedenkrát stisknete knoflík. Dále přesuňte šipku do políčka Datei na horním okraji obrazovky a znovu stisknete knoflík. Objeví se seznam příkazů. Šipka by měla být v poli Offnen. Pokud tomu tak není, přesuňte ji tam. Po stisku knoflíku se změní barva obrazovky na modrou a objeví se následující hlášení

Diskettenkopierprogramm

Zu kopierende Diskette in Laufwerk einlegen und Return drucken, um fortzufahren

Disketa, kterou chcete kopírovat se již nachází v dražvu, proto pouze stisknete Return. Přístroj se rozeběhne a objeví se hlášení

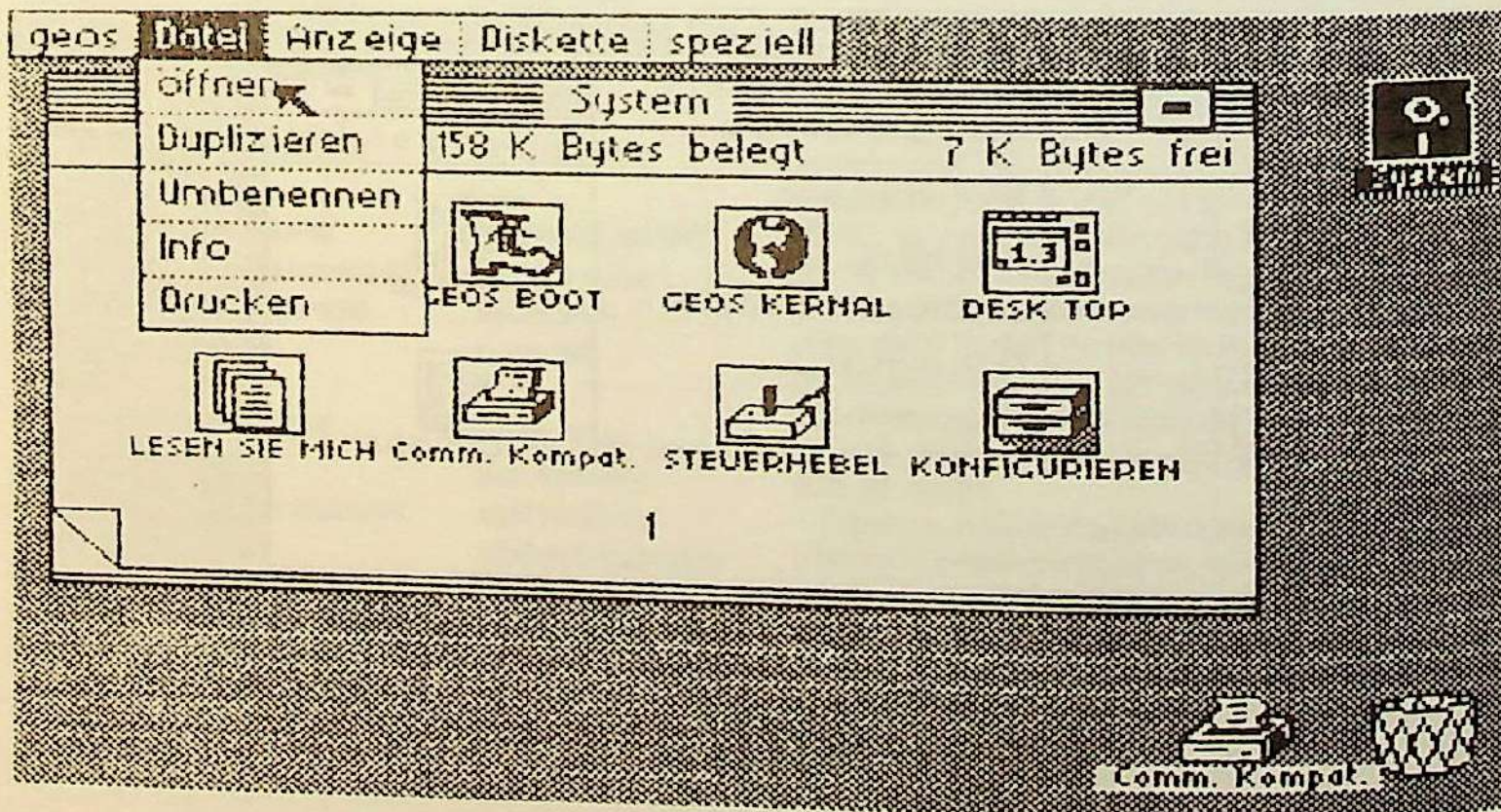
Liest Quelldiskette...

za okamžik se objeví následující sdělení
bitte einlegen

Zieldiskette

und Return drucken, um fortzufahren

Vyjměte zdrojovou disketu z přístroje a vyměňte ji za čistou. Tu vložte do dražvu a stisknete Return. Po aktivaci dražvu se objeví hlášení



Formatiert Zieldiskette...

o chvíli později se pak objeví následující text

Schreibt auf Zieldiskette...

Dále následuje sdělení

Bitte einlegen

Quelldiskette

und Return drücken, um fortzufahren

Vyjměte pracovní disketu z přístroje a nahraďte ji zdrojovou disketou s programy. Stiskněte Return. Dále postupujte podle výzev počítače a střídejte zdrojovou a cílovou disketu. Nakonec se objeví následující sdělení

Sicherungskopie fertig!

eine weitere Kopie anfertigen ?geben Sie J für ja und N für nein

Stiskněte N. Objeví se popis

B drücken, um zu Basic zurückzukehren,

oder Geos Ladediskette einlegen

und Restore drücken

Vyjměte z dráivu disketu Applications Disk a nahraďte ji disketou System. Stiskněte klávesu Restore popř. Run/stop Restore. Dráiv začne pracovat a po chvíli se opět na obrazovce objeví systémový deskTop.

Důležité upozornění! Backup System Disk chraňte na bezpečném místě. K ládování Geos ji používejte pouze tehdy, je-li poškozena systémová disketa. Applications Disk používejte jako zdrojovou pro zhotovení dalších pracovních kopií a k obnovení datových souborů, které se ztratily z vaší systémové diskety.

PÁR SLOV K TERMINOLOGII.

V dalších odstavcích budou definována slova a výrazy užívané v popisu systému Geos a jeho vlastností.

Menu, druhé menu, piktogram

Menu je přesně to, co si pod tímto pojmem představujete, totiž seznam funkcí. Menu může být použito k tomu, aby vás seznámilo s příkazy, pomocnými programy, druhy písma nebo libovolnými funkcemi. Menu příkazů na obrazovce deskTop kupř. seznamuje s funkcemi Geos, Datel, Anzeige, Diskette a Speziell. Je přístupné z deskTop.

Často vede volba určité funkce v menu k aktivaci dalšího menu, které se označuje jako menu druhé úrovně (druhé menu). Někdy vede volba určité funkce v druhém menu k otevření dalšího menu. Tato a další menu nemají potom speciální názvy. Zřídka se v Geos setkáte s více než dvouúrovňovým menu.

V bílém poli s označením Anwendungen, což představuje název diskety jsou údaje o počtu datových souborů a obsazených a volných bajtů. Pod nimi se nalézají malé symbolické obrázky, označované jako piktogramy. Symbol tiskárny a odpadkového koše v pravém dolním rohu je taktéž piktogramem. Tyto

obrázky reprezentují program nebo příkaz. Na listu představujícím seznam datových souborů jsou to obrázky deskTop, Diskettenkopier, geoPaint a geoWrite.

Komunikační okna

Často po vás program žádá určitou informaci a teprve po ní provede žádanou funkci. v Geos se tento kontakt děje pomocí komunikačních oken. Normálně se v tomto poli nachází sdělení, které uživatele upozorňuje, aby provedl volbu nebo zadal určitá data. Např. při volbě příkazu Umbenennen se objeví komunikační okno se sdělením

Bitte neuen Diskettenamen eingeben

V pravém dolním rohu se nachází obdélník s nápisem Abbruch. Abyste mohli s Geos pracovat dále, musíte si vybrat. Buď klávesou Del vymazat název diskety a napsat nový, nebo kurzorem najet na nápis Abbruch a stisknout knoflík. Zadaná data, resp. provedená volba je jako informace zpracovávána programem dále a komunikační okno zmizí.

Dokumenty

Aplikace geoWrite a geoPaint umožňují sestavení stránky 21,9 x 27,9 cm s vysokou rozlišitelností. Jednotlivé stránky se dají archivovat na disketě jako datové soubory. Pro jejich označení se vžil název dokumenty.

Okna

Orámovaný pravoúhlý útvar na obrazovce je všeobecně označován jako okno. Obvykle od sebe okna oddělují funkční rozsahy obrazovky. Např. geoPaint nabízí okno k malování pro dokument. V daném čase je na obrazovce pouze výřez z celého formátu. Dokument se dá v okně posouvat všemi směry, takže je dosažitelný každý jeho bod.

Druhy písma, způsob tisku, velikost bodů, proporcionalita

Druhy písma se určuje vzhled vašeho textu. Geos nabízí, stejně jako jiné komfortnější textové editory více znakových sad či druhů písma. Můžete psát středověkým typem písma Dwinelle nebo jednoduchým typem University. Výraz druh písma (font) označuje všeobecnou formu znaků, sloužících k zobrazení písmen a číslic. Datový soubor na disketě Geos může obsahovat znakové sady různé velikosti, jejíž jednotkou je jeden bod. Ve standardní sazečské terminologii je velikost bodu 0,35 mm. Ve skutečnosti však závisí velikost vytištěného druhu písma na rozlišovací schopnosti vaší tiskárny. Pro každý druh písma a velikost bodu se při tisku může měnit i jeho způsob. K dispozici je možno zvolit normal, kurzívu, tlusté, podtržené a obrysové písmo. Pokud si podrobně prohlédnete texty psané psacím strojem, zjistíte, že písmo je tištěno v přesně daných vzdálenostech. Porovnáme s pro-

fesionálně tištěnou stránkou, jako v tomto časopisu, uvidíte, že vzdálenost mezi písmeny závisí na jejich tvaru. „w“ je jistě širší než „l“. Závislost vzdálenosti mezi znaky na jejich šířce je potom nazývána proporcionality písma. Geos používá jen proporcionální písma, takže vaše dokumenty budou vyhlížet zcela profesionálně.

Pomocné programy

Jsou to malé služební programy, vyvolatelné z aplikací Geos. Po ukončení jejich práce se lze vrátit zpět do dané aplikace. Všechny pomocné programy se volají z menu Geos (v levém horním rohu). Jako příklad poslouží použití kalkulatoru v aplikaci geoWrite. Přijete pod geoWrite určitý text, ve kterém je třeba sečíst řadu čísel. V menu Geos aktivujete kalkulator, provedete nezbytné operace, kalkulator deaktivujete a pokračujete dále v psaní textu.

Klik, volba, aktivace

Většina příkazů v Geos se provede tím způsobem, že se ukáže na něco na obrazovce a stiskne se knoflík joysticku. Stisk knoflíku se označuje jako klik. Pokud přesunete šipku na určitou funkci a odkliknete, provedli jste určitým způsobem volbu. U některých piktogramů vede klik k provedení (aktivaci, otevření) funkce.

Uzavření diskety

po natažení Geos ze systémové diskety odkliknete slovo Diskette. Následně v menu, jež se nyní objeví

odkliknete slovo Schliesen. Seznam programů z tabule uprostřed obrazovky zmizí a piktogram diskety v pravém horním rohu nahradí otazník. Postup, který jste nyní provedli se označuje jako uzavírání diskety.

Otvírání diskety

Odkliknutím diskety s otazníkem se dá systémová disketa znovu otevřít. Vyzkoušejte si to. Na obrazovce se znovu objeví obsah systémové diskety.

Otvírání a uzavírání diskety jsou jedny z nejčastěji používaných operací v Geos.

Přejmenování diskety

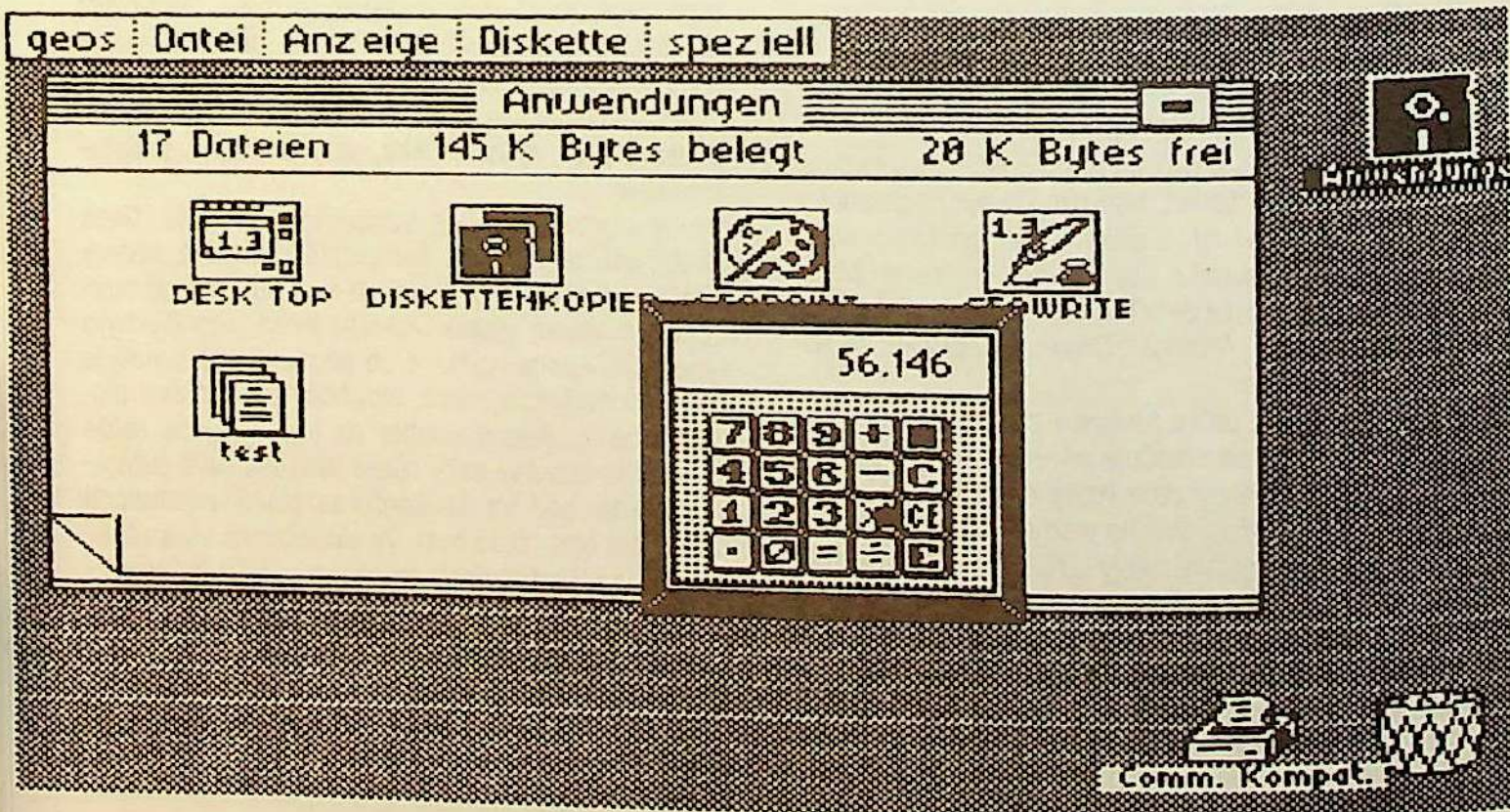
Aby se minimalizovaly přehmaty mezi dvěma disketami, pojmenujte každou disketu jinak. Odkliknete slovo Diskette v příkazovém menu. Potom vyberte slovo Umbenennen. Menu zmizí a objeví se komunikační okno s nápisem ve středu

Bitte neuen Diskettenamen eingeben
(název diskety)

Ve druhém řádku komunikačního okna se nachází aktuální název diskety. Na konci řádku potom najdete kurzor. Stiskněte klávesu INST/DEL a nakloňte páku joysticku ve směru vymazávání písmen. Už jste vymazali celý název? Nyní zadejte nový název, např.

Testy

a stiskněte Return. Komunikační okno zmizí. Znovu se objeví obsah diskety, přičemž bude akceptován nový název. Shodný název najdete i pod piktogramem diskety v pravém horním rohu.



Přízpusobení pracovní diskety

Abyste si na pracovní disketě udělali více místa pro vlastní práci, je nezbytné vymazat některé datové soubory.

Upozornění! Mazání datových souborů je ošidná záležitost. Jednou způsobená škoda se těžko nahrazuje. Ujistěte se proto nejdříve, že máte z dané diskety dostatek kopií, teprve potom začněte mazat!

Změna ochrany proti přepsání u individuálního datového souboru

Mnohé datové soubory z Geos jsou chráněny proti přepsání, je tedy vyloučena možnost je při nešetrné manipulaci nějak poškodit. Protože potřebujeme některé datové soubory přesto vymazat, musíme nejprve tuto softwarovou zádržku odstranit. Z diskety, kterou budete používat pro psaní textů můžete vymazat např. geoPaint. Nejprve odklikněte piktogram geoPaint. Piktogram se objeví v inverzním zobrazení. Potom odklikněte funkci Datei a v menu, které se objeví pak funkci Info. Objeví se okno s řadou informací. Pro nás je důležitý čtvereček vlevo dole. Vedle něj je potom slovo Schreibgeschützt. Nyní lze lehce rozpoznat, zda se datový soubor dá vymazat či nikoli. Pokud je čtvereček vyplněn černě, datový soubor se nedá z diskety vymazat. Prázdný čtvereček potom označuje soubor, který není ochráněn před přepsáním.

Datový soubor geoPaint se dodává s ochranou proti přepsání. Pro její zrušení odklikněte jednoduše černý čtvereček. Černá výplň zmizí. Opětovným klikem se objeví zpět, což znamená, že soubor je opět chráněn. Pokaždé, když toto pole odkliknete, změní se stav ochrany proti přepsání.

Předtím než budete pokračovat, ujistěte se, že jste ochranu zrušili. Nyní uzavřete okno. Odklikněte bod, který se nachází v pravém horním rohu okna. Po provedeném úkonu okno zmizí, přičemž datový soubor není nadále chráněn.

Vymazání datového souboru

Pracovní disketa je nyní připravena tak, aby z ní mohl být bez obtíží vymazán datový soubor geoPaint. Odklikněte příslušný piktogram, chvíli počkejte a přidejte ještě jeden klik. Objeví se stínový piktogram, který je možno pomocí šipky posunovat po obrazovce. Odtáhněte piktogram k odpadkovému koši a tam klikněte. Piktogram krátce zabliká. Potom zmizí stínový i reálný piktogram. Změna se přenesla do lišty na horním okraji, kde jsou uvedeny informace o počtu volných a obsazených bajtů.

Listování obsahem diskety

Na 5 1/4. disketě pro mechaniku VC 1541 se dá uchovávat až 144 různých datových souborů. Originální disketa s Geos je dodávána s 26 soubory. Jak jste si nepochybně všimli, je na jedné stránce obsahu

diskety místo právě pro 8 datových souborů. Aby bylo vidět ostatní piktogramy, je nutno obrátit stránku. Levý dolní roh stránky je jakoby ohnut. Pokud jej odkliknete, objeví se další stránka. Tímto způsobem můžete prohlédnout celý obsah diskety. Klik na čistou plochu pod ohnutým rohem vede k listování v opačném směru. Přímý skok na určitou stránku je možný tehdy, pokud zadáte číslo stránky. Uděláte-li chybu a zadáte číslo vyšší než je skutečný počet všech stran obsahu, nic se nestane. Program bude špatný údaj ignorovat.

Kopírování datových souborů z jedné diskety na druhou

V předchozích odstavcích byl popsán způsob kopírování celé diskety pomocí kopírovacího programu Diskettenkopier. Někdy je však nezbytné bezproblémově okopírovat z jedné diskety na druhou právě jeden datový soubor. Vyskytne se kupř. požadavek okopírovat na pracovní disketu budič pro tiskárnu. Prvním krokem je otevření zdrojové diskety. Potom následuje nalistování stránky s požadovaným budičem tiskárny. Odklikněte příslušný piktogram, chvíli vyčkejte a klikněte podruhé. Nyní vytáhněte stínový piktogram ven pod okno a tam jej odložte. Uzavřete disketu. Pozorujete, že piktogram zůstal tam, kam byl odložen? Vymějte z dráhy zdrojovou disketu a vložte do ní disketu, na niž má být program překopírován.

Upozornění! Nezkoušejte překopírovat datový soubor, pokud již datový soubor shodného jména existuje. Stane-li se tak jeden z datových souborů přejmenujte. Nyní odklikněte piktogram budiče tiskárny vně listu a za chvíli klikněte podruhé. Potom vtáhněte piktogram dovnitř a odložte jej na vhodné místo. Přenesení piktogramu dovnitř okna interpretuje Geos jako kopírování. Objeví se komunikační okno, které vás vyzve k založení datové diskety se souborem. Potom Geos přečte takovou část datového souboru, která se vejde do bufferu pro kopírování. Potom se objeví jiné komunikační okno vyzývající k založení cílové diskety. U dlouhého datového souboru postup zopakujete několikrát. Po ukončení kopírování se piktogram s budičem tiskárny přenesla do okna.

GEOWRITE

Najděte stránku diskety na níž se nachází geoWrite. Pro natažení programu do počítače použijte zkráceného postupu, kdy šipkou najedte na daný piktogram a dvakrát stisknete. Po úspěšném dvojitém kliku začne dráhu pracovat. Pokud se snad geoWrite na obrazovce neobjeví, bude to mít následující příčiny:

1. Při velmi rychlém kliknutí se často rozliší pouze jeden klik, datový soubor je pak pouze aktivován.
2. Při pomalých klicích se objeví stínový piktogram, který je posunovatelný šipkou.

V obou případech posuňte šipku jen o kousek vedle

původního piktogramu a klikněte. Datový soubor se dostane zpět do neaktivního stavu. Předpokládejme však, že váš pokus byl úspěšný a geoWrite je natažen do počítače. Na obrazovce se potom objeví komunikační okno v němž je možno zvolit sestavení nového dokumentu, otevření starého či návrat zpět do deskTop.

Odklikněte slovo Erstellen. Objeví se nové komunikační okno, které se bude dotazovat na název dokumentu. Zadejte libovolný název (Dopis). Komunikační okno zmizí a na obrazovce bude pouze okno pro psaní textu. Na horním okraji okna je lišta s povely, které jsou rozděleny do šesti skupin: Geos, Datei, Editieren, Optionen, Schriftart a Stil. Pokud odkliknete některou funkci, objeví se menu, ze kterého můžete volit. Vpravo od této lišty je ukazatel stran. Vedle se nacházející šipky slouží k rolování stránkou o jeden řádek nahoru či dolů, podle toho kterou šipku odkliknete. Dále vedle vpravo se nachází lišta s titulem tvořeného dokumentu. Přímo pod povelovou lištou je umístěno pravítko. Pomocí něj můžete změnit formát svého dokumentu. Uvnitř okna se potom nalézá známá šipka a textový kurzor, pod kterým se objevuje klávesnicí zadaný text. Pro naládování geoWrite existuje vedle dvojitěho kliku ještě jedna možnost. Najdete a odkliknete piktogram geoWrite a potom v menu políčka Datei zvolíte Offnen. Který způsob zvolíte, záleží pouze na vás.

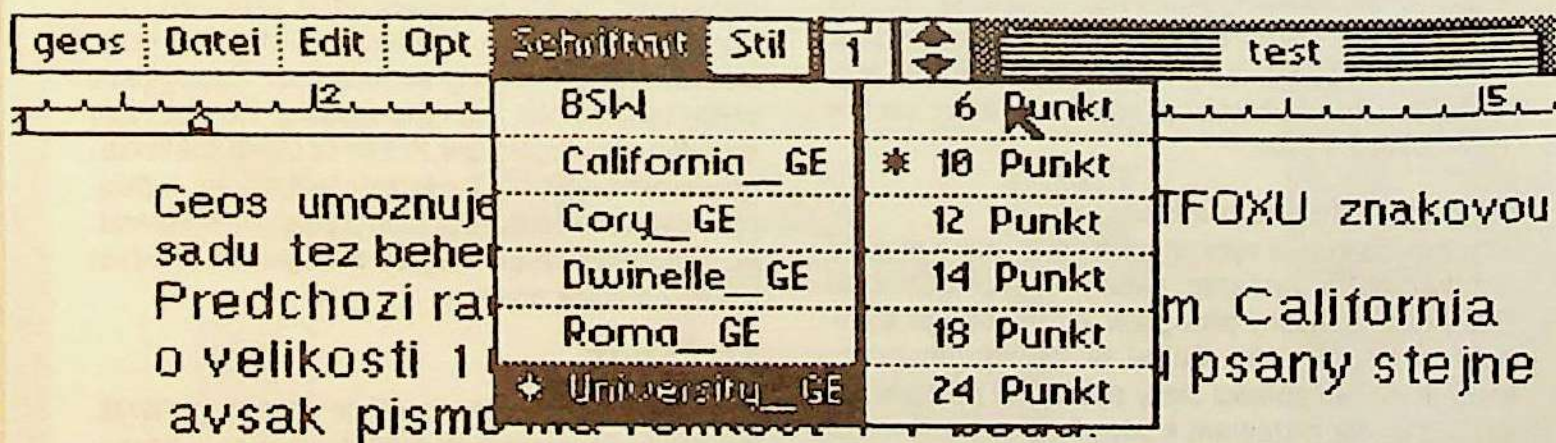
Funkcí Existierendes Dokument offnen otevřete dokument na disketě. Objeví se komunikační okno s aktuálním názvem diskety a lišta s názvy prvních pěti dokumentů. Další názvy se objeví po každém kliku na rolo-

vací šipku na spodním okraji okna. Vyberte příslušný dokument a odklikněte políčko Offnen.

Dokument se dá otevřít i z deskTop. K tomu účelu aktivujte piktogram dokumentu a příkaz Offnen z menu Datei nebo piktogram dokumentu dvojitě odklikněte. Oběma postupy se automaticky dotáhne geoWrite a otevře se váš dokument.

Vytváření textu

Text se do dokumentu vnáší jednoduše přes klávesnici. Na pozici textového kurzoru se objeví písmeno stisknuté klávesy. Pozici textového kurzoru lze změnit tím způsobem, že šipkou najedete na určité místo na obrazovce a odkliknete. Textový kurzor se přenes. Při psaní vznikají překlapy. Opravovat je můžete tak, jak jste běžně zvyklí, klávesou DEL. konec odstavce následuje po stisku Return. Textový kurzor se tím přesune automaticky na začátek dalšího řádku. GeoWrite umožňuje použití 8 různých tabulátorů. Pro každou pozici tabulátoru existuje na pravítku zvláštní značka. Tabulátor se nastaví tak, že se najede šipkou na značku tabulátoru. Klikem se značka přilepí k šipce, takže ji lze snadno přesunout do požadovaného místa. Dalším klikem se potom značka odloží. Pro vymazání pozice tabulátoru odtáhněte značku do horní poloviny pravítka a tam klikněte. Při psaní textu se na jednotlivé tabulátory skáče stiskem kláves CTRL a I. Na levé a pravé straně pravítka se vedle značek tabulátorů nacházejí ještě dvě značky jimiž jsou určovány okraje textu. Značky mají podobu písmene M. Jejich polohování se provádí stejným způsobem jako polohování tabuláto-



Comotronic spol. s r.o.

Dolnomlynska 2

Sumperk psc 787 01

rů. Začátek nové textové stránky zvolíte v menu Option odkliknutím slova Seitenanfang. Protože okno zobrazuje jenom část textu na stránce, dochází během práce často k jeho přesouvání. Prvá možnost přesouvání vzniká jako vedlejší produkt psaní textu. Buď je ukázána levá nebo pravá polovina textové stránky, podle toho, kde zrovna píšete. Rovněž tak rolování textu nahoru působí zcela automaticky. Mimo to lze rolovat nahoru a dolů pomocí šipek vedle ukazatele čísla stránky. Tento ukazatel má ještě jednu funkci. Všimli jste si malého obdélníčku uvnitř ukazatele. Jeho aktivací se lze pohybovat po celé stránce. Pro přechod z jedné strany na druhou zvolte v menu Option funkce nachste Seite nebo vorige Seite. Příímý skok na požadovanou stránku se provede odkliknutím funkce Seite auswahlen a zadáním čísla stránky.

Druhy písma, styl tisku

V geoWrite můžete kdykoll změnit druh a velikost písma. Zvolte v poli Schriftart druh písma a začněte psát změna může být provedena kdykoll. Příímou volbou lze určit i styl tisku (kurzíva, tlusté, obrysové či podtržené písmo).

Edítace

Vložení textu je stejně jednoduché jako zadávání textu. Šipku nastavte do místa kam má být vložen text a odklikněte. Na stejné místo se přestaví i textový kurzor. Potom můžete začít s vkládáním textu. Při použití klávesy DEL můžete naopak přebytečný či jinak závadný text vymazat.

Tisk

Vyberte a okopírujte příslušný budič tiskárny. Zapněte tiskárnu a v menu Datei zvolte příkaz Drucken.

(JV)

SID - SYMPHONY

(přečetli jsme za Vás)

Ještě před několika roky byl C 64 se svým tříhlasým syntetizérem senzací. Technika však nezůstala stát. Moderní počítače mají hlasů mnohem více. Také C 64 má nyní možnost počet svých hlasů znásobit.

Americká firma CMD uvedla na trh modul SID – Symphony. Tím jistě potěšila řadu nadšenců, kterým jeden SID v C 64 byl málo. Modul totiž obsahuje další SID.

Připojení modulu zvládne každý. Nasunuje se do expanzního portu. Z konektoru CINCH na boku pouzdra modulu je možno druhým tříhlasým zvukem napájet jeden kanál stereofonního zařízení. Druhý kanál lze spojit s audiovýstupem C 64.

Po mechanickém propojení se může přikročit k naládování programu. U některých typů přístrojů (C 64, VC 1541) mohou zklamat urychlovací loudry. Naštěstí je možné použít standardní nahrávací rutiny. Pracují spolehlivě. V malém okénku se objeví zvuky nahrané ve formě dat na disketě. Výběr se provádí kurzorem. Stiskem na F1 se dá vstoupit do následující obrazovky. C 64 natáhne požadovaný

kus a ihned začne s jeho reprodukcí. Na klávesnici, která se objeví, se vybarví klávesy, které se stisknou. Vedle klávesnice je zobrazen pomocný displej, který ve stylizované formě ukazuje úroveň všech šesti hlasů. Přehrávání zvuku přes stereoaparaturu je skutečně impozantní. Z počátku se ani nechce věřit, že C 64 může produkovat tak fantastický zvuk.

Prvý software umožňoval pouhé přehrávání uložených skladeb. To bylo velkým záporem při prvním seznámení s modulem. Nyní se však s modulem dodává další software, který pomůže i nemuzikálním typům vyvíjet své vlastní šestihlasé skladby. Pro tento účel se do počítače naláduje stereoeditor. Zvuk se dá jednoduše komponovat přes klávesnici či joystickem. Funkčními klávesami lze vstoupit do dalšího menu. Vedle síly a tempa je možno navolit zpoždění a další parametry. K zadávání textu slouží vestavěný Word – Prozessor. Paralelně ke zvuku se dá potom na obrazovce zobrazit i text.

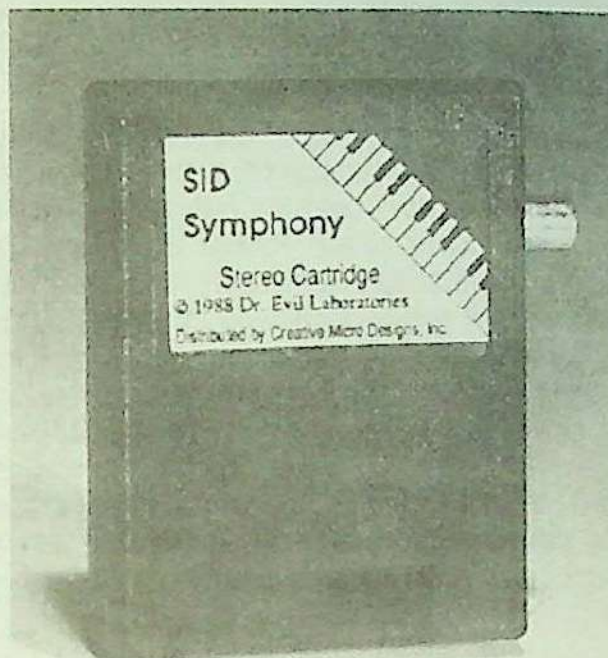
Mnohostrannost možností, které program nabízí, se ocení teprve po delším používání. Žel, doku-

mentace SIDu sestává z pěti oxeroxovaných kopií. Vlastní návod je napsán na disketě, komplet v angličtině. Muzikanti však, doufejme, nebudou mít problémy, neboť program se dá ovládat z grafických menu. Každý ze šesti hlasů se dá jednotlivě naprogramovat. Nechybí samozřejmě možnost kompletně celou kreaci přehrát a potom uložit na disketu.

ZÁVĚR

S novým software se hodí modul k zábavě. Optimální zvuk se projeví teprve ve spojení se stereozařízením. Použitím druhého Sidu se dosáhne vynikajícího oddělení kanálů. Záporem je nedostatečná průvodní dokumentace.

(JV)



DOPISY ČTENÁŘŮ

TIPY A TRIKY OD ČTENÁŘŮ

Záchrana programu

po nechtěném NEW nebo RESET lze program zachránit následující sekvencí příkazů:

POKE 2050,1:SYS 42291:POKE 46,PEEK(35):POKE 45,PEEK(781)+2:CLR

bude-li systém hlásit chybu, je nutno odeslat:

POKE46,PEEK(35):POKE 45,PEEK(781)-254

Znemožnění RUN/STOP + RESTORE

POKE808,225

– znemožnění použití RUN/STOP + RESTORE k přerušení programu

POKE808,237

– povolení použití RUN/STOP + RESTORE

PRINTPEEK(17)

=0 poslední vstup dat pomocí INPUT

=64 poslední vstup dat pomocí GET

=152 poslední vstup dat pomocí READ

(Martin Štěpnička, Přelouč)

Další jednořádkové pouky pocházejí od jiného autora.

Rychlost blikání kurzoru ovlivní

POKE 56325,5 – maximální rychlost

POKE 56325,255 – minimální rychlost

POKE 56325,50 – normální rychlost

Vypnutí kurzoru způsobí zadání

POKE 788,123

jeho opětovné zapnutí potom

POKE 788,49

Pomalý tisk znaků na obrazovku

POKE 56325,0

Modifikovat se dá i listing

POKE 774,0 – vytiskne jen čísla řádek

POKE 774,27 – uvede vše jako zkratky

POKE 774,26 – normální listing

POKE 775,1 – zákaz listingu

POKE 775,167 – povolení listingu

(J. Dušička ml., Úpice)

READY nebo HOTOV?

Anglická hlášení, která vypisuje C64 na obrazovku lze změnit na hlášení česká. Jako příklad uvádím změnu hlášení READY na HOTOV. Opište do počítače následující program a spusťte jej.

```
10 FOR X=40960 TO 49151: POKE X,PEEK (X): NEXT:
POKE 1,54
20 FOR X=41848 TO 41853: POKE X,32: NEXT
30 A$="HOTOV.": FOR X=1 TO 6: POKE 41847+X,
ASC(MID$(A$,X,1)): NEXT
```

Jak program funguje? V řádku 10 se interpreter Basic přenesou z paměti ROM do RAM. V řádku 20 se smaže hlášení READY a v řádku 30 se potom zapíše do paměti hlášení HOTOV. Zbývá jen podotknout, že majitelé FCIII musejí před spuštěním programu modul odpojit příkazem KILL.

Jak vyvolat smazaný program ?

Program, který byl smazán příkazem NEW, SYS64738 nebo RESET tlačítkem nelze již spustit, přestože se dosud nachází v paměti počítače. Následujícím trikem jej můžeme vyvolat zpět:

```
POKE 2049,1: POKE 2050,1: SYS 42291
```

Plynoucí text

```
100 PRINT CHR$(147)
110 A=39
120 GOSUB 230
130 P$=LEFT$(T$(X),A)
140 T$(X)=RIGHT$(T$(X),LEN(T$(X))-A)
150 POKE 214,23: POKE 211,0: SYS 58732
160 PRINTP$
170 P$=RIGHT$(P$,A-1)+LEFT$(T$(X),1)
180 T$(X)=RIGHT$(T$(X),LEN(T$(X))-1)
190 IF LEN(T$(X))=0 THEN X=X+1
200 IF T$(X)="KONEC" THEN GOSUB 230
210 GOTO 150
220 :
230 RESTORE: X=0
240 READ T$(X)
250 IF T$(X)="KONEC" THEN 280
260 X=X+1
270 GOTO 240
280 X=0
290 RETURN
300 :
310 DATA "TENTO PROGRAM SLOUZI
K PREDSTAVENI TEXTU NA OBRAZOVCE"
320 DATA "KONEC"
```

Program na obrazovce vykouzlí efekt, při kterém písmenka běží z pravé strany na levou. Umístění plynoucího textu na obrazovce lze změnit v řádku 150. Délka řádku, ve kterém text běží je dána hodnotou proměnné A. Do řádků s daty lze umístit libovolně dlouhý text, který musí být vždy zakončen slovem KONEC.

(D. Šutera, Litovel)

PRINTFOX – diakritika u velkých písmen

Program Printfox je jeden z nejlepších DTP programů na C64. Rozšířená česká verze obsahuje sice českou diakritiku, jedná se však pouze o malá písmena. Nedostatek volných kláves (některé klávesy jsou obsazeny příkazy) způsobuje potíže při použití české diakritiky u velkých písmen. Tato situace má několik řešení:

1. Vyhnout se použití těchto znaků přesunem slov. Výhodou je, že potřeba velkých písmen vůbec nevzniká. Nevýhodná je někdy nutnost používat doslova krkolomné slovní obraty, kdy je jazyk výslovně przněn.
2. Domalovat diakritická znaménka do textu přímo v grafice.

Výhoda je shodná jako v předchozím případě. Nevýhodou je zbytečné zdržování při opakovaném tisku. Pomoci si můžete uložením celých „českých“ grafických obrazovek na disketu. Vznikají však další zvýšené nároky na místo na disketě.

3. Upravit znakovou sadu změnou málo používaných znaků pomocí programu CHARACTERFOX. V mém případě to byly znaky << a >>. Místo těchto znaků jsem nadefinoval čárku nad písmeno a háček, neboť tyto znaky lze pak kdykoli dosadit do textu pomocí zpětného posuvu a subscriptu a při opakovaném tisku jste bez problémů. Nevýhodou je vinou řídících znaků v editoru méně čitelný text. Při zarovnávání na oba okraje je nutno používat různý zpětný posuv. Jako příklad uvádím: HÁČKY A ČÁRKY Předchozí text vytvoříme tak, že napíšeme C<4^(háček)^(space)<4A. Aby bylo vše úplně jasné podotýkám, že znak < předpokládá současný stisk kláves (CTRL) a (šipka vlevo), zatímco znak ^ vyjadřuje subscript, tedy stisk kláves (CTRL) a (šipka nahoru).

Na závěr ještě poznámka pro ty, kteří potřebují psát číselné tabulky (i hexadecimální) a nechť se vzdá grafiky. Při úpravě znakových sad se doporučuje změnit šířku číslic a znaků A až F na stejnou velikost (např. 5 bodů). Číslo lze psát potom přímo do sloupců.

(SGQ, Ostrava)

SCHEMAFOX

Protože jsem radioamatér, nashromáždilo se mi za léta množství schémat a různých časopisů, např. Amatérské rádio, Sdělovací technika apod., ve kterých jsou opět uveřejňována různá schémata. Katastrofální ovšem je, když potřebuji nějaké to schéma rychle najít. Kde to a to schéma bylo? Prohlížet stohy časopisů nebo ručně kreslených schémat je časově velmi náročné. Papír taktéž zabírá hodně místa. Bez úspěchu jsem si zkoušel dělat různé seznamy na disketách. PRINTFOX mne přivedl na správnou myšlenku. Vytvořil jsem doplněk k tomuto programu a nazval jej SCHEMAFOX. Pomocí něj lze snadno kreslit schémata různých elektrotechnických zařízení, ukládat je na disketu a případně je nechat vytisknout na tiskárně. Program zabírá zhruba jednu polovinu běžné 5 1/4" diskety. Obsahuje zpracovaný návod, titulek diskety, vzorek schématu. Podle složitosti se potom vejde na stranu diskety 5 1/4" asi 50 až 60 schémat.

(Jiří Placer, Veronské nám. 331, 10900 Praha)

Vytváření slovníků

Basic V2.0 umožňuje sestavit různé typy výpočetních, zábavných i vzdělávacích programů. Jím může být i slovník, umožňující zábavnou formou procvičení slovíček cizího jazyka za pomoci počítače. Poměrně dokonalých a obsáhlých slovníků pro C64 je nabízena řada v různých inzercích. Následující příspěvek nenabízí listing programu, ale popisuje některé zásady a náměty pro sestavení vlastního slovníkového programu. Podle uvážení si pak může každý, kdo se zajímá o programování a využívání programů sám takový program stvořit.

Každý slovníkový program se skládá z ovládací části a slovní zásoby. Do slovníku potřebujeme slovíčka zapisovat a opravovat je. Poměrně jednoduše lze tento požadavek v Basicu ošetřit tak, že slovíčka se zapisují v datové části programu např.:

```
1000 DATA HOSTEL=KOLEJ, STOVE=KAMNA,  
      BEYOND=ZA, TYRE=PNEUMATIKA  
1001 DATA HOUSEHOLD=DOMÁCNOST,  
      ITEM=POLOŽKA, POLITE=ZDVOŘILÝ
```

Důležité je uvedení oddělovacího znaménka =, které rozdělí výraz na část anglickou a českou. V tomto případě bylo zvoleno rovnítko, nicméně klávesnice C64 poskytuje celou řadu dalších možností. Podle oddělovače bude program rozeznávat při zkoušení českou a anglickou část.

V ovládacím programu má hlavní úlohu výběr a čtení slovíček a zkoušení. Slovíčka budeme do slovníku opísat pravděpodobně z některé učebnice, kde jsou uspořádána po lekcích nebo abecedně. Je proto třeba vybrat slovíčka ke zkoušení náhodnou volbou. Tomuto účelu poslouží následující podprogram:

```
10 REM NAHODNA VOLBA  
20 X=TI  
30 L=RND(X)  
40 I=INT(SZ*RND(L))+1  
50 RETURN
```

Před použitím funkce RND je vhodné zadat řádek 20, kdy se pro náhodnou volbu vytváří nové výchozí číslo, jinak budou generovány stejné sledy náhodných čísel. V řádku 40 je SZ počet slovíček ve slovníku, I je pak vypočtené pořadové číslo slovíčka pro příkaz READ, kterým je slovíčko načteno z datové části.

Pro vlastní zkoušení lze vzít za základ např. tento podprogram:

```
10 REM ZKOUŠENÍ SLOVÍČEK  
20 PRINTSHIFT-CLR:PRINT:PRINT  
30 READ SL$(I):R$=SL$(I)  
40 FOR J=1 TO LEN (R$)-1  
50 A$=LEFT$(R$,J)  
60 IF RIGHT$(A$,1)=“,“ THEN GOTO 80  
70 NEXT J:  
80 PRINT TAB(4);A$;  
90 AT=TI  
100 GET W$: IF W$=“,“ THEN GOTO 100  
110 U=(TI-AT)/60  
120 INPUT C$  
130 PRINT TAB (23);U;“,SEC“:PRINT  
140 IF A$=C$><R$ THEN PRINT  
      TAB(7);“,CHYBA“;SPC(5);R$  
150 RETURN
```

Podprogram přečte slovíčko, podle oddělovače nalezneme levou část, tedy v zápisu CALL=VOLAT se na obrazovce objeví levá část CALL. Pokud víme význam, stiskneme některou klávesu (řádek 100), potom je vypočten čas reakce v sekundách a můžeme napsat význam VOLAT. V případě, že jsme napsali význam špatně, objeví se hlášení chyba a současně se na obrazovce objeví správný výraz, v našem případě CALL=VOLAT. Máme-li takto zapsané slovíčko v datové části, uvedeným podprogramem budeme zkoušeni ze zpětného překladu českých výrazů k anglickým. Při hledání anglických významů k českým je potřeba, aby program přečetl výraz CALL=VOLAT obráceně, nebo před použitím podprogramu ZKOUŠENÍ SLOVÍČEK je třeba slovníkový výraz obrátit na tvar VOLAT=CALL.

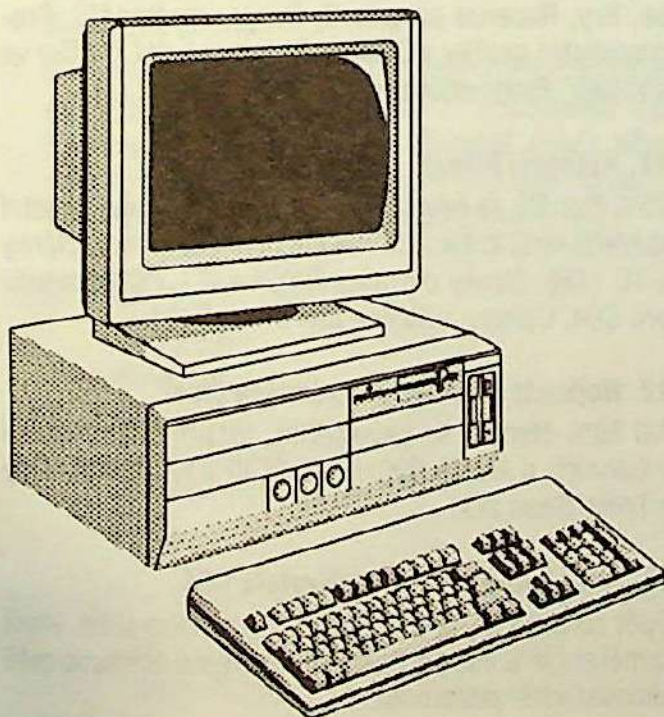
Pro tento účel může posloužit podprogram:

```
10 REM PŘEVŘÁCENÍ ŘETĚZCE
20 READ SL$(I):R$=SL$(I)
30 FOR J=1 TO LEN (R$)-1
40 A$=LEFT$(R$,J)
50 IF RIGHT$(A$,1)=,," THEN GOTO 70
60 NEXT J70 A$=LEFT$(A$,LEN(A$)-1)
80 FOR J=1 TO LEN (R$)-1
90 B$=RIGHT$(R$,J)
100 IF LEFT$(B$,1)=,," THEN GOTO 120
110 NEXT J120 B$=RIGHT$(B$,LEN(B$)-1):F$=,,"
130 R$=B$+F$+A$
140 RETURN
```

Další funkcí, kterou může slovník plnit, je sečtení slovíček zapsaných ve slovníku. Operace se dá provést např. takto:

```
10 REM SOUČET SLOVÍČEK
20 FOR I=1 TO 10 000
30 READ SL$(I)
40 IF SL$(I) = ,,"+++“ THEN GOTO 60
50 NEXT I60 N=I-1: PRINT SPC(5),,SLOVNÍK OBSAHUJE";N;,,"SLOV“
70 RETURN
```

Pro tento účel musí být ve slovníku jako poslední řetězec napsána zvolená posloupnost znaků, kupř. +++.



Další úlohou programu je načtení slovíček do proměnné a seřazení podle abecedy:

```
10 REM ABECEDNÍ ŘAZENÍ
20 FOR I=1 TO N30 READ SL$(I)
40 NEXT I50 PRZN = 1
60 IF PRZN=0 THEN PRINT ,,SERAZENI UKONCENO“
70 PRZN = 0
80 FOR I=1 TO N-1
90 IF SL$(I) = SL$(I+1) THEN GOTO 140
100 P$=SL$(I)
110 SL$(I)=SL$(I+1)
120 SL$(I+1)=P$
130 PRZN=1
140 NEXT I150 GOTO 60
170 RETURN
```

Slovíčka seřazená podle abecedy nebo načtená přímo ze slovníku lze po stránkách prohlížet na obrazovce, resp. mohou být vytištěna tiskárnou. Jednu stránku zvolíme např. 20 slovíček. Předpokládáme, že slovník je načten do proměnné SL\$(I) a počet slovíček uveden v proměnné N

```
10 REM ČTENÍ STRÁNKY SLOVNÍKU
20 Z=INT(INT/20)+1
30 INPUT ,,VLOŽTE ČÍSLO STRÁNKY“ ;CS
40 J=(CS-1)*20
50 PRINT SPC(2); Z; SPC(10); CS
60 FOR I=J TO J+20
70 PRINT SPC(2), SL$(I)
80 NEXT I90 GET W$: IF W$=,," THEN 90
100 IF W$=CRSR DOWN THEN 140
110 IF W$=CRSR UP THEN 130
120 IF W$=,,"R“ THEN RETURN
130 CS=CS-1: GOTO 40
140 CS=CS+1: GOTO 40
```

CRSR DOW NCRSR UP jsou kódy kláves kurzor dolů a kurzor nahoru. Touto volbou lze vyvolat následující či předcházející stránku slovníku, tedy prolístovat celý slovník. Úprava podprogramu pro tisk je již jen rutinní záležitostí.

Pokud chcete vybavit ovládací část programu čtením stránek jen s anglickými nebo jen s českými výrazy, je potřeba každý výraz rozdělit na anglickou a českou část a ty uložit do řetězcových proměnných, z nichž je bude možno vyvolat nebo nechat vytisknout. Další možností si lze vymyslet celou řadu.

(Č. Florián, Chrudlm)

VYHODNOCENÍ ČTENÁŘSKÉ SOUTĚŽE

Vážení čtenáři, musím na začátku konstatovat, že tolik chvály jsme si od Vás nezasloužili a tolik spokojenosti jsme také neočekávali. Z ankety jsme si chtěli vzít ponaučení, jak zlepšit naši práci, kterým směrem se dále zaměřit, abyste byli s časopisem ještě spokojenější. Jenže – posuďte sami.....

NEJDŘÍVE, KDO VYHRÁL:

I. cenu, FCIII přiklepla ruka štěstěny Jiřímu Placerovi z Prahy 10.

II. cenu, joystick COMPETITION PRO získal Marek Mička z Krnova.

III. cenu, joystick SV 124 jsme zaslali výherci, ing. Košťovi do Košic.

Výhercům blahopřejeme a věříme, že jim výhry přinesou užitek.

ROZBOR ODPOVĚDÍ

1. Rozsah časopisu

90 procent účastníků soutěže odpovědělo, že rozsah je vyhovující, ostatní by byli raději, kdyby měl časopis více stránek, 30 – 48 a to i ve formátu A4.

2. Cena časopisu

Všichni považovali cenu za přijatelnou a ti, kteří by rádi větší počet stran, souhlasili s vyšší cenou.

3. Obsah časopisu

Obsah časopisu považuje za výborný asi 30% čtenářů, vyhovující 70%. Pouze jeden čtenář uvedl, že časopis neodpovídá jeho potřebám.

4. Rubrika Basic

Rubrika Basic je většinou považována za vyhovující, je zde však cca 25% čtenářů, kteří ji považují za příliš jednoduchou nebo odbornou.

5. Rubrika Assembler

Rubrika Assembler je rozporuplnější, než Basic. Je to dáno tím, že vyžaduje více přemýšlení. Výsledkem je, že 70% uživatelů je s rubrikou spokojeno, 25% ji považuje za příliš odbornou a 5% za příliš jednoduchou.

6 – 7. Představujeme Vám a Comotronic News

Tyto rubriky považujete všichni za užitečné.

8. Nejhorší rubrika

Za nejhorší rubriku považuje 5% respondentů Assembler, 3% články o grafice a 2% rubriku Basic. Ostatní se buď nevyjádřili nebo považují všechny rubriky za užitečné.

9. Nejlepší rubrika

Asi 25% čtenářů neví, 4% považují za nejlepší BASIC, 6% Assembler a Představujeme Vám, zbytek našel zálibení v rubrice Tipy a Triky.

10. Jakou novou rubriku doporučujete

Protože se v této rubrice plně projevilo, že každý se zabývá něčím jiným, bylo by nejvhodnější zavést rubriku „Od všeho trošku“. Požadujete totiž rubriky Hudba, Hry, Recenze programů, Programy čtenářů, Programování grafiky v Basicu, Programování grafiky ve strojáku, Programování RS 232 atd., atd.

11. Nejhorší článek

75% čtenářů se nevyjádřilo, 15% čtenářů neví, ostatní rozdělili svoji kritiku zhruba stejným dílem mezi články o TC 16kB, články o grafice DYCP a FLI, PC klávesnici pro C64, Opravy a Změna cen dovozů atd.

12. Nejlepší článek z 5. vydaných čísel.

Asi 80% čtenářů se nevyjádřilo, ostatní našli zálibení v článcích o Amize 600/1200, FCIII a ty ostatní, Tipy a Triky, Geos atd.

13. Vaše zaměření jako uživatele C64

Opět co uživatel, to jiný názor. Proto nelze určit, které zaměření je u Vás nejčastější a kterému bychom měli věnovat větší pozornost.

14. Co v oblasti HW nejvíce postrádáte

Postrádáte vše. Od levných disketových mechanik, přes monitory, po (samozřejmě levné) RAM, EPROM moduly a interfejsy. Hlavní problém tak není, že daný HW není na trhu, ale že stojí většinou víc peněz, než dokáže naše ekonomickou reformou vyčerpaná peněženka vydat.

15. Co postrádáte v oblasti SW

Postrádáte nejrůznější programy. Většinou by Vás nadchly české mutace známých programů jako PRINTFOX, VIDEOFOX, EDDISON, SUPERBASE, GEOS v cenové relaci do 100,- Kč. Přesto Vaším odpovědím na tento bod budeme věnovat mimořádnou pozornost.

16. Nejbližší prodejna Commodore ve vašem okolí
50% Vás nezná jinou prodejnu než Comotronic Šumperk. Rádi bychom Vám pomohli tím, že bychom uveřejnili seznam prodejen, které my známe, ale velký počet prodejen v letošním roce mění sortiment a odchází od počítačů Commodore, takže to rychle není možné. Přesto považujeme, jak Vám v této oblasti vyjít vstříc.

A JAK JE TO JINDE?

Shodou okolností se mi dostaly do rukou výsledky čtenářské ankety, pořádané německým časopisem 64'er. Zde jsou její hlavní výsledky:

- téměř 1/2 respondentů vlastní C64 více než 3 roky,
- 38% respondentů se počítá mezi začátečníky, s pouze malými znalostmi, asi 60% se považuje za pokročilé a pouze kolem 2% uživatelů o sobě tvrdí, že jsou profíci.
- asi 95% uživatelů používá disketovou mechaniku, asi 55% Datassette. Přibližně 77% uživatelů má tiskárnu. Mezi HW, který si uživatelé nejvíc přejí patří tiskárna a monitor. Výsledky odpovědí na otázku, obdobné našim jsou také zajímavé:

nejžádanější rubriky časopisu:

Tipy a triky	- 67%
Novinky	- 63%
Programování	- 53%
Řešení problémů	- 52%
Soukromé využití	- 52%
GEOS	- 47%

nejčtenější rubriky

Tipy a triky	- 93%
Testy software	- 70%

Aktuality	- 69%
Softwarová pomoc	- 64%
Testy hardware	- 62%

Méně čtené jsou části, zabývající se hrami. Asi 20% je nečte vůbec nebo zřídka, ale asi 35% pravidelně. Téměř polovina dává svůj výtisk časopisu číst ještě někomu dalšímu.

Zajímavá je také odpověď na otázku „co děláte se svým počítačem, co vás zajímá“. Dále uvedená procenta se vztahují na odpověď „více než 1 hod. týdně“:

Zpracování textu	- 54%
Hry	- 53%
Programování	- 48%
Grafika, malování	- 38%

Na konci seznamu jsou pak měření, řízení, regulace, Teletext a přenos dat telefonním modemem. Zajímavé je také zjištění, že 37% tvrdí, že hry přestávají být středem jejich zájmu.

A ještě věkové složení čtenářů:

Průměrný věk 28 let, mladších 20.let je 40%, nejstarší čtenář má 77 let.

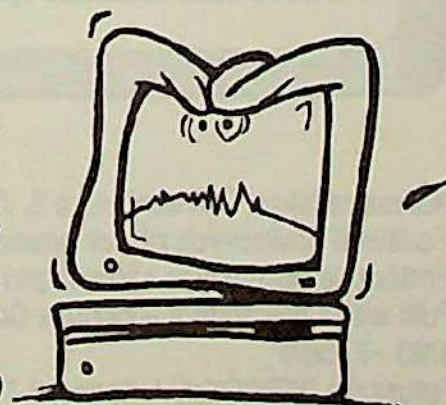
CO ŘÍCI ZÁVĚREM?

Když nic jiného, utvrdila nás čtenářská soutěž v tom, že naše práce je užitečná a časopis Vám něco dává. Pokud bychom chtěli vyvodit závěry z Vašich připomínek, měli bychom se zaměřit na to, abyste měli dostatek dobrých a levných programů. Malý krůček v tomto směru představují naše programové kazety/diskety s návodem. Věříme, že jejich řada se do budoucna rozšíří.

Srovnání s německou anketou kupodivu ukazuje přibližně stejné zájmy čtenářů, ačkoliv materiální podmínky a doba intenzivního využívání počítačů C64 jsou s našimi nesrovnatelné.

Vaše připomínky a náměty rozhodně nezapadnou, budeme se k nim vracet při přípravě dalších článků a čísel FUnu. Proto závěrem děkuji všem, kteří si dali tu námahu a odpověděli na naši anketu.

(JK)



INZERCE

Predám program Anglicko-Slovenský slovník na C64. Obsahuje vyše 10 000 anglických slov a ich prekladov, dokáže povedať ľubovoľné anglické slovo, možno v ňom listovať, zmesť sa na necelú disketu a stojí 50,- Sk.

Stanislav KUCHAR
Vajanského 1129/29
024 01 Kysucké Nové Mesto

Predám počítač COMMODORE 128 s magnetofónom COMMANDER za 6980,- Sk, FINAL CARTRIDGE za 850,- Sk. K tomu zdarma časopisy COMMODORE 64'er, manuály a 3 JOYSTICKY

Emil ŠVEDA
SNP 1482-137/11
017 01 Považská Bystrica
tel. 0822/62351

Sháním kódy do zemí ke hře Bards Tale III. Vyměním za cokoli jiného, příp. jinou literaturu (např. kouzla k Elvíře). Platí stále!

Marek Václavík
tř. 1. máje 1621
753 01 Hranice

Předplatné časopisu FUN with Commodore

Cena za jedno číslo v předplatném na minimálně 5 čísel je 10,- Kč. Ročník 1993 bude mít celkem 10 čísel. V současné době máme ještě v omezeném množství k dispozici všechna vyšší čísla, která je možno dokoupit za cenu 5,- Kč za číslo z ročníku 1992 a 15,- Kč za číslo z ročníku 1993.

Předplatné se platí složenkou na adresu firmy Comotronic, na záda poukázky pro příjemce napište **FUN 93 – počet čísel**. Kopii nebo druhý kontrolní ústřížek zašlete společně s objednávkou na adresu firmy Comotronic. Pokud jste zaplatili i starší čísla, budou vám odeslána společně s dalším novým číslem, které vyjde.

Adresa redakce: Dolnomlýnská 2, 787 01 Šumperk * Autoři čísla: Jaroslav Vančura a Jiří Kouřil
Podávání novinových zásilek povoleno Oblastní správou pošt v Ostravě pod č.j. 2882/92-P1 ze dne 14.12.1992
Podávání novinových zásilek povoleno SP š.p. ZsRP Bratislava č.j. 613-PD-1993 ze dne 1. apríla 1993
FUN with Commodore – časopis Comotronic klubu pro uživatele počítačů Commodore 64/128.
4/93 – 7. číslo
Fotosazba REPROtisk J. Kotinský * Tisk: Vegaprint Šumperk



Vydává Comotronic klub Šumperk