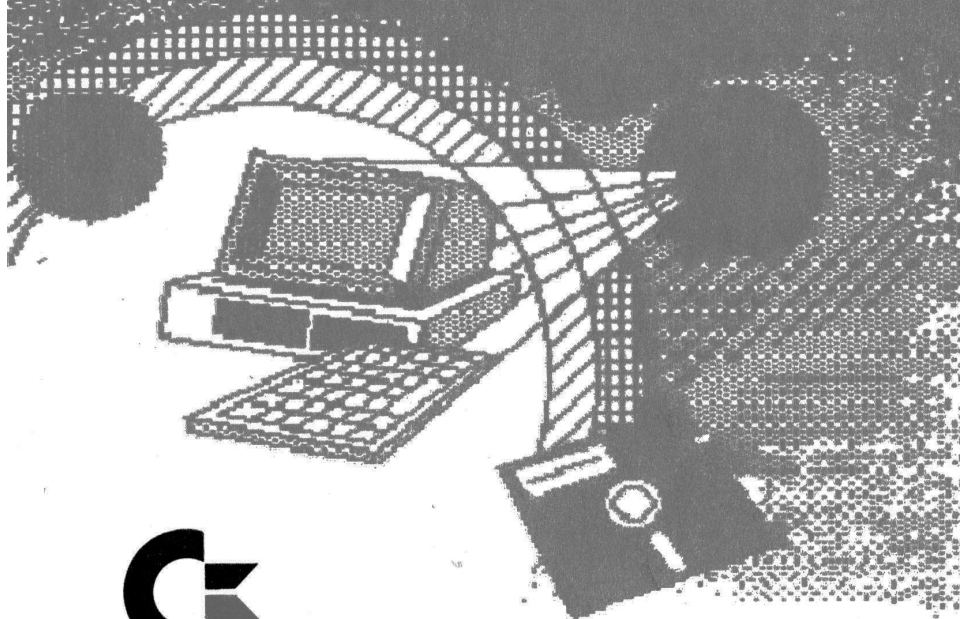


Commodore Klub



KRAJSKÁ SEKCIA PRI KDPM KOŠICE

SPRAVODAJ

3

Číselné sústavy

(František KUPEC)

Číselné sústavy sprevádzajú človeka od nepamäti. Pri ich zrode zohrala zrejme úlohu potreba evidovať množstvo rôznych predmetov, materiálov a pod. Tak sa postupne vyvinul už dnes viac-menej dokonalý systém vyjadrenia číselných hodnôt prostredníctvom určitých dohodnutých znakov – **číslic**.

Číselné sústavy, teda systémy na vyjadrenie počtu, delíme na dva typy – **pozičné** a **nepozičné**. Nepozičné sa už dnes používajú veľmi zriedka. Patrí medzi ne napr. rímska sústava. Pozičnou sústavou je zaužívaná desiatková. Jej vznik bol zrejme podmienený počtom prstov na oboch rukách a teda najvhovujúcejším počtom spravidla v každej situácii. Keďže táto sústava vyhovovala najviac, ľudstvo sa s ňou zžilo a nevznikala potreba vytvárať iné typy číselných sústav. S rozvojom výpočtovej techniky však vznikla požiadavka zaviesť inú než desiatkovú sústavu. V súčasnosti sa najviac využíva dvojková, osmičková a šestnástková, ktoré taktiež prináležia k sústavám pozičným.

Všetky pozičné sústavy sa vyznačujú tým, že celková hodnota, ktorú predstavuje určitý číselný znak, závisí nielen od **hodnoty čísla**, ale aj od jeho umiestnenia v rámci čísla. Poloha číselného znaku v rámci čísla sa nazýva **rád**, ktorému v pozičných sústavách zodpovedá príslušná **váha**. Jej hodnotu určíme ako mocninu základu číselnej sústavy, pričom exponent tejto mocniny je určený poradím daného rádu (sprava 1, 2 atď.).

Výslednú hodnotu čísla vyjadrenú pomocou číselných znakov určíme súčinom jednotlivých cifier a k nim príslúchajúcich pozičných váh.

Napríklad číslo 1358 zapíšeme ako súčet:

$$1358 = 1 \times 10^3 + 3 \times 10^2 + 5 \times 10^1 + 8 \times 10^0$$

V desiatkovej sústave sú teda váhami čísla 10^0 (jednotky), 10^1 (desiatky), 10^2 (stovky), 10^3 (tisíciky) atď.

Dvojková sústava – používa na vyjadrenie počtu len dva znaky 0 a 1. Tieto dva znaky, ktorým môžeme priradiť len

dva stavy, sú v technickej praxi ľahko realizovateľné. Medzi ďalšie výhody patrí jednoduchý prevod do často používanej osmičkovej a šestnástkovej sústavy. Avšak nevýhodou je, že čísla takto vyjadrené majú veľký počet číslic. Jednotlivé rády tu predstavujú čísla $1=2^0$, $2=2^1$, $4=2^2$, $8=2^3$, $16=2^4$, atď.

Šestnástková sústava (hexadecimálna) – tejto sústave už nestačí bežne používaných 10 číslic (0 až 9) desiatkovej sústavy, preto sa na vyjadrenie chýbajúcich 6 číslic používajú číselné znaky A, B, C, D, E, F. Pozičné váhy tejto sústavy sú $1=16^0$, $16=16^1$, $256=16^2$, atď.

Prevod z desiatkovej do dvojkovej sústavy.

1. spôsob – metóda delením a zapisovaním zvyškov. Pri tomto prevode delíme prevádzané číslo základom sústavy, do ktorej prevádzame (2) s použitím aritmetickej sústavy, v ktorej sa číslo nachádza pred prevodom. Po prvom delení dostávame číslo, ktoré opäť delíme základom sústavy a tento postup opakujeme dovtedy, kým výsledkom čiastkového delenia nie je 0. Zvyšky jednotlivých čiastkových delení zapisujeme. Dvojkový ekvivalent prevádzaného čísla získame opísaním zvyškov delení v opačnom poradí, ako sme ich získali (t. j. sprava doľava).

Napr.:

53	26	13	6	3	1	0	– výsledky delení
2	1	0	1	0	1	1	– zvyšky

$(53)_{10} = (110101)_2$

2. spôsob – metóda odčítania pozičných váh. Od prevádzaného čísla postupne odpočítavame vždy najbližšiu menšiu, alebo rovnú pozičnú váhu a pokračujeme dovtedy, kým nedostaneme 0. Samotný dvojkový ekvivalent získame tak, že do príslušného rádu zapíšeme 1, ak sme jeho pozičnú váhu odpočítali od prevádzaného čísla. 0 zapíšeme v prípade, že daná pozičná váha je väčšia ako číslo, od ktorého by sme ju mali odčítať.

Napr.:

$53 - 1 \times 32 - 1 \times 16 - 0 \times 8 - 1 \times 4 - 0 \times 2 - 1 \times 1 = 0$

Prevod z dvojkovej do desiatkovej sústavy

1. spôsob – desiatkový ekvivalent získame sčítaním súčinov cifier a im prislúchajúcich váh, ktoré sú vyjadrené v desiatkovej sústave.

Napr.:

$$(10101)_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = \\ = 16 + 0 + 4 + 0 + 1 = 21$$

2. spôsob – pomocou Hornerovej schémy

$$\begin{array}{r} \text{I} \\ \times z \quad \text{II} \\ \text{III} \end{array}$$

z – základ číselnej sústavy, v ktorej je vyjadrené prevádzané číslo (t. j. 2)

I. riadok – cifry prevádzaného čísla

II. riadok – medzivýsledky násobené

III. riadok – výsledky sčítania čísel I. a II. riadku

Pri prevode postupujeme tak, že najprv spíšeme cifru najvyššieho rádu z I. riadku do III. riadku. Ďalej vynásobíme číslo z III. riadku 2 a výsledok tohto násobenia zapíšeme do II. riadku v stĺpci pod druhým najvyšším rádom. Obidve čísla v tomto stĺpci sčítame a výsledok zapíšeme pod nich do III. riadku. Tento postup opakujeme, až kým nedostaneme desiatkový ekvivalent sčítaním čísel v poslednom stĺpci.

Napr.:

$$\begin{array}{r} 1 \quad 0 \quad 1 \quad 0 \quad 1 \\ 2 \quad \quad 2 \quad 4 \quad 10 \quad 20 \\ 1 \quad 2 \quad 5 \quad 10 \quad 21 \end{array}$$

$$(10101)_2 = (21)_{10}$$

Prevod z desiatkovej do šestnástkovej sústavy

Výhodným sa javí analogické použitie metódy dele-

ním a zapisovaním zvyškov ako u dvojkovej sústavy, pritom však pochopiteľne delíme základom tejto sústavy, teda číslom 16. Odlišnosťou je zapisovanie zvyškov väčších ako 9 v šestnástkovom tvare (t. j. číslice A, B, C, D, E, F).

Napr.:

$$\begin{array}{r} 167 \quad 10 \quad 1 \\ 16 \quad 7 \quad 10 = A \\ (167)_{10} = (A7)_{16} \end{array}$$

Prevod zo šestnástkovej do desiatkovej sústavy

Opäť postupujeme podobne ako v prípade s dvojkovou sústavou a z oboch spôsobov prevodu sa zdá byť výhodnejším postup, pri ktorom sčítavame súčiny šestnástkových cifier a im prislúchajúcich pozičných váh šestnástkovej sústavy.

Napr.:

$$\begin{aligned} (A7)_{16} &= (A)_{16} \times 16^1 + (7)_{16} \times 16^0 = \\ &= 10 \times 16 + 7 \times 1 = 160 + 7 = (167)_{10} \end{aligned}$$

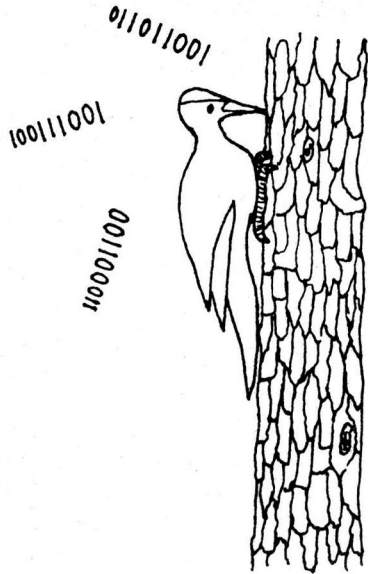
Tipy a triky

(Pavol ŽALOBÍN)

```

10 INPUT" OBSAH DISKU V BASIC ";Q$
12 IF Q$="A" THEN 1( 1
13 END: : : : : : : : : : : : : : : : : : : : : : : : : : :
101 OPEN 1,8,0,"$":GET#1,A$,A$: E$=CHR$(0)
102 GET#1,A$,A$,H$,L$:IF ST THEN CLOSE 1: RETURN
103 PRINT "<LEFT>"ASC(H$+E$)+256*ASC(L$+E$);
104 GET#1,A$,B$:IF A$ THEN PRINT A$B$;:GOTO 104
105 PRINT A$: GOTO 102

```



Obmedzenie klaviatúry

```

5  REM Obmedzenie pre čísla 1-20
10 INPUT A$
20 IF ASC(A$) < 48 OR ASC(A$) > 57 THEN 55
30 A = VAL(A$)
40 IF A < 2 OR A > 20 THEN 55
45 GOTO 60
55 PRINT"<2xUP>": GOTO 10
60 : : : : : : : : : : : : : : : :

```

RESTORE

```

POKE 792,193 – vypnúť
POKE 792, 71 – zaonúť

```

SAVE

```

POKE 819,246 – vypnúť
POKE 819,245 – zapnúť

```

LOAD

```

POKE 816,157 – vypnúť
POKE 816,165 – zapnúť

```

PROFI-ASS 64

(Igor KOLLA)

PROFI-ASS je tvorený v strojovom kóde a má dĺžku 4 kB (4096 byte). Je umiestnený od adresy 36864, čiže sa musí nahráť príkazom **LOAD"PROFI-ASS 64",8,1**. Keďže ukazovateľ BASICu sa preniesie na koniec programu t. j. 40960, je potrebné zadať po nahratí príkaz NEW, inak vypisuje chybu OUT OF MEMORY a nedá sa v ňom pracovať. Programátor má k dispozícii 34 kB pamäte. PROFI-ASS 64 používa štandardný BASIC-ovský editor, takže je nenáročný na obsluhu. Inštrukcie ASSEMBLERU sa zadávajú podobne ako BASIC-ovské príkazy s číslami riadkov.

Riadky sa skladajú z návestí, mnemotechnických inštrukcií, operandov a komentáru.

10 START LDA #10 ; komentár

V Profi-Ass sa môže dokonca niekoľko ASS inštrukcií písať do jedného riadku (cez dvojbodku).

20 LDX #10: STX \$D020

Existujú tu aj pseudoinštrukcie, ktoré nepredstavujú žiadne strojové inštrukcie, ale prikazujú previesť špeciálne úkony. Každému riadku s inštrukciou môžeme priradiť aj návestie (znakový symbol) o dĺžke 9 znakov. Návestie sa píše pred inštrukciu a je oddelené jednou alebo viacerými medzerami. Začína písmenom a musí sa odlišovať v prvých ôsmych znakoch. Iné, ako alfanumerické znaky nie sú povolené. Mnemotechnické inštrukcie nie je možné použiť ako návestie.

Ak začína inštrukcia ".", ide o pseudoinštrukciu. Všetky pseudoinštrukcie musia byť od svojich operandov oddelené medzerou. Ak sa riadok ukončí ";", nasleduje komentár. Pole operandu obsahuje spôsob adresovania a výraz pre inštrukciu.

Spôsoby adresovania majú nasledujúcu syntax:

Inštrukcia	Adresovanie	
LDA	#N	priame adresovanie
LDA	N	adresovanie na nultú stránku
LDA	N,X	na nultú stránku + X (Y)
LDA	NN	absolútne adresovanie
LDA	NN,X	absolútne adresovanie + X (Y)
LDA	(N,X)	
LDA	(N),Y	nepriame adresovania
JMP	(NN)	

Poznámka: N predstavuje 8-bitové číslo.

VÝRAZY

PROFI-ASS 64 je mnohostranný, pretože je schopný vypočítať ľubovoľné komplikované výrazy. Je povolené aj používanie hranatých a okrúhlych zátvoriek.

K dispozícii sú tieto operátory:

+	sčíta hodnoty
-	odčíta pravú od ľavej
*	násobenie
!	logické OR
?	logické XOR

Operandy sa môžu vyskytnúť vo viacerých formách.

Typ	Syntax
Hexadecimal	\$1BA5
Decimal	53280
Binar	%10110101
ASCII znak	"a"

PROFI-ASS 64 je uložený od 36864-40960 (\$9000-\$A000). Restart robíme povelom SYS 36864.

PROFI-MON 64

(Igor KOLLA)

PROFI-MON 64 je uložený od adresy 49152-51200 (\$C000-\$C800). Zavádza sa ako LOAD"PROFI-MON 64",8,1. Po natiahnutí je potrebné zadať povel NEW. Štartovacia adresa je 49152, takže po zadaní SYS 49152 sa na obrazovke vypíše obsah všetkých registrov.

	PC	IRQ	SR	AC	XR	YR	SP	NV	BDI	ZC
>;	1968	EA31	34	39	32	35	F8	00	11	0100

Monitor slúži na úpravu, editáciu, prehľadávanie atď. programov, ktoré sú v strojovom kóde. Má štandardne ovládateľný editor, podobný ako BASIC. Príkazy monitora sú jednopísmenové a po nich zvyčajne nasleduje adresa (START, END).

Profi-Mon 64 umožňuje nasledujúce funkcie:

- D** Adr1 Adr2
[DUMP] – urobí disassembler programu od adresy Adr1 po Adr2
- M** Adr1 Adr2
[MEM] – vypíše oblasť pamäte od adresy Adr1 po Adr2
- H** Adr1 Adr2 Byt Byt Byt . . .
[HUNT] – hľadá od adresy Adr1 po Adr2 za sebou idúce byty – Byt, Byt . . .
Adresy nájdených bytov vypíše
- F** Adr1 Adr2 Adr3
[COMPARE] – porovná oblasť pamäte od adresy Adr1 po Adr2 s adresou Adr3 až Adr3+(Adr2-Adr1)
- R** [REG] – Vypíše obsahy všetkých registrov
- G** Adr
[GO] – štart programu od adresy Adr

- X** prechod do BASICu
- L** "názov", číslo prístroja, Adr1
 [LOAD] – nahrá program do počítača z daného prístroja
 01 – Datasette
 08 – Disk
 od adresy Adr1
- S** "názov", číslo prístroja, Adr1, Adr2
 [SAVE] – nahrá program (oblasť pamäte) na Datasette alebo Disk.
 Nahrá sa oblasť od adresy Adr1 po Adr2

DEBUGGER 64

(Igor KOLLA)

Debugger 64 je tvorený v strojovom kóde a má dĺžku 4 kB. Je umiestnený od adresy 28672, takže sa musí nahráť príkazom **LOAD "DEBUGGER 64", 8, 1**. Po nahratí je potrebný príkaz **NEW**. Debugger 64 slúži na odladovanie programov, písaných v strojovom kóde.

Umožňuje nasledujúce funkcie:

- S** vykoná jednu inštrukciu
- T** [TRASE] – automatické vykonávanie príkazu **S**
- E** rýchle vykonávanie inštrukcií bez výpisu registrov
- P** rýchle vykonávanie inštrukcií s výpisom registrov
- J** [JSR] – vykoná naraz celý podprogram
- B** [BEEP] ON/OFF – zapína/vypína pípanie
- G** [GO] – štart programu od adresy PC
- R** [REG] – zmena registrov
- P** nastaví PC (program counter)
- A** nastaví A (acumulator)
- X** nastaví X (index register X)
- Y** nastaví Y (index register Y)
- S** nastaví SP (stack pointer)
- F** nastaví S (status register)
- M** [MEM] – vypíše oblasť pamäte od adresy PC
- D** [DUMP] – urobí disassembler programu od adresy PC
- N** [NOP] – nevykoná nasledujúcu inštrukciu
- W** [WINDOW] – vytvorenie grafického okna pri grafických módoch

COLOSSUS 2

(Milan BOBULA)

Colossus patrí k najkvalitnejším logickým a šachovým programom vytvoreným na mikropočítačoch Commodore a Atari. Jeho herné kvality sú výsledkom niekoľkoročnej práce na zdokonaľovaní programu. V súčasnosti existujú v páskovej a diskovej verzii Colossus 2 a 4 vytvorené P. B. Bryanom. V štvorke sú uplatnené nové, komfortnejšie možnosti, predovšetkým s akcentom na riešenie šachových problémov. Vzhľadom na väčšie rozšírenie Colossus šachu 2 sme sa zamerali práve na spracovanie tohoto návodu. Treba poznamenať, že jednotlivé verzie majú špecifické odlišnosti a z toho dôvodu nie je možné v plnej miere manuál aplikovať u verzii 4.



— OPĀŤ SI BOL ZA AMIGOU!

Základné informácie

Po načítaní celého programu (v diskovej verzii **LOAD"CHESS",8,1**) a jeho štarte je na monitore zobrazaná šachovnica so základným pozičným rozostavením figúr. Zároveň sa objaví hlásenie "YOUR MOVE". Teraz je program pripravený pre vašu voľbu. K dispozícii máte biele kamene a môžete začať hru, prípadne zmeniť základné parametre definované programom.

Zmena zobrazených farieb

Po stlačení klávesy C môžete meniť farby v nasledujúcom poradí:

INK COLOUR mení svetlé polia, resp. farbu figúr
PAPER COLOUR mení farbu pozadia a tmavých polí
BORDER COLOUR mení farbu rámu obrazu

Jednotlivé zmeny robíme klávesami CRSR a SHIFT v rozmedzí farebnej škály 0–15. Hlásenie ILLEGAL sa objaví v prípade, že INK a PAPER majú navolenú rovnakú hodnotu farby a musíte urobiť zmenu.

Nastavenie typu a obtiažnosti hry

Základné nastavenie hracej úrovne v móde 2 je 10 sekúnd na uvažovaný ťah. Ak chcete obtiažnosť korigovať, stlačte klávesu M a 2x RETURN. Na obrazovke sa objaví MINS 00. Podobným spôsobom, ako u korekcie farby (pomocou CRSR) navoľte požadovaný čas pre prácu programu. K dispozícii je rozsah 0–59 minút na ťah. Po stlačení RETURN-u podobne postupujeme pri nastavení času v sekundách. Pre bežného hráča zvyčajne postačuje úroveň do hodnoty 1–5 minút.

Význam jednotlivých módov

Po stlačení M je v základnom nastavení inicializovaný MODE 2. K dispozícii je však 6 charakteristických typov hry.

MODE 1 – praktizuje sa pri turnajoch. Nastavujú sa jednotlivé časové kontroly.

MODE 2 – predstavuje obtiažnosť hry. Čím viac času poskytnete programu na uvažovanie, tým bude jeho hra presnejšia a kvalitnejšia.

MODE 3 – praktizuje sa často ako blicka, šach s ukončením do určitého času. Každý hráč má pevne stanovenú dobu, za ktorú musí previesť všetky kroky do ukončenia partie. Hra končí matom, patom alebo padnutím času u niektorého zo súperov.

MODE 4 – V tomto móde sa počítač snaží robiť ťahy v približne rovnakom časovom intervale ako vy. Tempo je regulované podľa rýchlosti vašej hry.

MODE 5 – Šach sa snaží nájsť najhodnotnejšie a najideálnejšie ťahy tak, aby sa zbytočne nezaoberal bráním figur, ak to nie je nevyhnutne potrebné. Snaží sa čo najrýchlejšie ukončiť partiu, alebo nájsť najvýhodnejšiu pozíciu.

MODE 6 – Používa sa pri riešení a realizácii špecifických šachových problémov, pričom hľadá najoptimálnejší počet ťahov na ukončenie partie.

Šachová notácia

Pred začatím hry, prípadne v jej priebehu máte možnosť sledovať prácu programu. Stlačením medzery SPACE sú následne zobrazené nasledujúce informácie:

- ubehnutý čas od začiatku hry u oboch hráčov
- časť šachovej notácie (zapamätaných je až 120 ťahov, teda aj tie, ktoré nie sú momentálne zobrazené)
- LOOKAHEAD, hĺbka dopredu prepočítaných ťahov
- POSITION EXAMINED, počet kalkulovaných pozícií
- BEST LINE, stav hry. MTRL porovnáva „stav materiálu“. Ak stojí horšie Colossus, je pred udávanou hodnotou znak -. PSQL udáva kvalitu pozície
- V spodnej časti obrazovky sú zobrazené najvýhodnejšie ťahy, ku ktorým program dospel.

Nastavenie plynúceho času

Po stlačení klávesy E môžeme u oboch strán nastaviť hodnotu šachových hodín. Všimnite si, že pri akomkoľvek zásahu do programu je plynutie času zastavené. V hernom móde 3 môžete počítaču prípadne sebe skrátiť čas na uvažovanie, čím docielite následnú zmenu hernej úrovne a tempa.

Šachová hra

Po požadovanom nastavení parametrov prístupíte k samotnej hre. Pomocou kurzorových kláves premiestnite kurzor na pole s figúrou, ktorou chcete ťahať. Stlačte RETURN. Teraz presuňte kurzor na výsledné pole a opäť stlačte RETURN. Podobný efekt dosiahnete algebraickým zápisom, ale musíte držať stlačenú klávesu SHIFT. Pokiaľ ste neporušili šachové pravidlá, ťah je prevedený a zároveň zaevidovaný v notácii. V dolnej časti obrazovky sa zobrazí hlásenie "LET ME THINK" – nechaj ma premýšľať. Program teraz prepočítava najvýhodnejší protiťah. Stlačením klávesy CBM môžete jeho činnosť kedykoľvek prerušiť.

Ovplyvnenie hry

Zadanie B: Po stlačení klávesy B bude prevedený ná-

vrat realizovaných ťahov.

Zadanie F : Používa sa pri analýze partie. Prevádza ťah dopredu, pokiaľ bol vrátený klávesou B, alebo bol realizovaný.

Zadanie N : Po súčasnom stlačení kláves CTRL a N zabezpečíte základné rozostavenie figúr a môžete začať novú partiu. Zároveň sú vynulované šachové hodiny.

Zadanie P : Po stlačení tejto klávesy bude hrať počítač za obe strany až do prerušenia pomocou CBM, prípadne do ukončenia.

Zadanie G : Po tomto zadani na začiatku partie začne hru Colossus ako prvý za biele kamene.

Špeciálne možnosti

Zápis, prípadne nahratie uloženej partie:

Stlačte klávesu T pričom sa na obrazovke objaví hlásenie LOAD OR SAVE. Zadaťte S a zabezpečíte si tak uloženie rozostavenej partie na mgf. kazetu. Ak zadáte L, môžete si partiu kedykoľvek nahráť do pamäte a neskôr v nej pokračovať. Clonený šach dosiahneme po stlačení klávesy I. Táto varianta sa praktizuje u veľmi dobrých hráčov. Program je v režime "INVISIBILITY – neviditeľnosti". Opätovným stlačením I sa obraz uvedie do pôvodného stavu. Príkaz O zabezpečí otočenie šachovnice. Zadanie V mení hlasitosť zvukovej indikácie v rozsahu 0–15.

Výučba šachu

Prejdite na pole s ľubovoľnou figúrou a stlačte klávesu L. Všetky legálne ťahy, ktoré môže zvolená figúra previesť sú indikované blikajúcim kurzorom. Ak figúra nemá možnosť ťahu, ostane kurzor na pôvodne navolenom poli.

Výmena pešiaka za inú figúru

Pokiaľ sa vám počas hry podarilo doviest pešiaka na pole premeny (1 respektíve 8 riadok šachovnice), objaví sa hlásenie "PROMONTE TO? – povýšiť na?". Program žiada hodnotu figúry za akú ho mienite zameniť. Na zadanie slúžia tieto klávesy: N – jazdec, B – strelec, R – veža, Q – dáma.

Roscháda

V prípade, že chcete uskutočniť roschádu, je potrebné

presunúť kráľa o dve políčka doprava, prípadne doľava (podľa toho, na akú stranu chcete roschovať). Veža sa automaticky presunie na správnu pozíciu. Samozrejme, že tento ťah sa dá uskutočniť len ak neboli porušené šachové pravidlá.

Branie ent passant

Branie mimochodom je v notácii označené ako EP. Krok je na šachovnici vykonaný a inicializovaný blikajúcim kurzorom.

Colossus šach disponuje ďalšími možnosťami, súvisiacimi s analýzou partí, navolením herných problémov a pod., ktorým sa budeme bližšie venovať v ďalšom čísle Spravidaja.

EGA - grafický editor

(Ing. Ján ZAIC, Pavol ŽALOBÍN)

Odborný časopis užívateľov zariadení firmy Commodore, 64-ER v marci 1988 uverejnil program EGA. Vzhľadom na vynikajúce vlastnosti tohoto grafického editora sme sa rozhodli uverejniť návod na jeho obsluhu. Program pracuje v minimálnej konfigurácii C-64, VC1541/1571/1581, MPS801/803/1000/Epson, Joystick/1531/No1.

EGA – nová dimenzia v kreslení

EGA je kresliaci program novej generácie. Môžeme ním naraz spracovávať štyri stránky obrazovky. Programový balík obsahuje 3 základné programy EGA V3.2, EGA-PRINT.GEN, ZEICHENSATZ. Po natihnutí EGA-PRINT.GEN (tlačová rutina) vytvorí sa program EGA – PRINT, uložený od \$0400. Hlavný program EGA V3.2 sa naťahuje ako **LOAD"EGA V3.2",8** a štartuje povelom RUN.

Informatívna obrazovka

Po natihnutí programu a štarte sa na obrazovke objavia súradnice: x-vľavo hore a y-vľavo dole. V strede horného rámu obrazovky vidieť skrinku, delenú do štyroch častí. Vždy je invertovaná tá časť, ktorej je priradená aktuálna HiRes obrazovka. EGA pracuje s kresliacou plochou 640×400 bodov. Máme

k dispozícii 4 obrazovky. V strede spodného rámu obrazovky vidieť žltý obdĺžnik, ležiaci nad nápisom **SPEED**. Tým sa označuje variabilná rýchlosť kurzora, ktorá je na začiatku programu nastavená na maximum (obdĺžnik má celú dĺžku). Pod obdĺžnikom sú tri písmená, označujúce aktuálnu kresliacu funkciu. Hneď po štarte je zvolená funkcia DRW. V pravom spodnom rohu je v jednom sprite zobrazená vzorka pre napĺňanie (vedľa slova **FILL**). Posledný je sprite v pravom hornom rohu. Tento sprite obsahuje **M** a dve čiarky, znázorňujúce uhol, pod ktorým sa pomocou kurzora presúvajú preberané výseky grafiky príkazom MOVE. Výrez je uzavretý práve medzi obidvoma čiarami. Program sa riadi joystickom (myšou) z portu 2.

Príkazy sa členia na hlavné a vedľajšie. Označené sú len hlavné príkazy. Všetky funkcie (až na ZOOM) sa prerušia pomocou klávesy «STOP». Potom je automaticky aktivovaný príkaz DRW. Pretože u funkcií ako LIN je nasadený ďalší kurzor, je program stavaný tak, že pri zmene obrazovky nie je kurzor viditeľný, avšak pri návrate do východzej obrazovky sa opäť objaví.

Ak funkciu nemožno vykonať, alebo bolo vykonané chybné zadanie, vydá program hlboký tón ako chybové hlásenie. Pri niektorých príkazoch potvrdí program správne zadanie vysokým tónom. Skoro všetky kresliace funkcie možno vykonávať na všetkých štyroch obrazovkách.

Hlavné príkazy

«SPACE» **ZOM**: ZOOM (zväčšiť). Aktivuje režim zväčšovania, tzn. výrez 40×25 pixelov sa zobrazí zväčšene. Sprite ukazujúci aktuálnu obrazovku zmizne a aktivuje sa až potom, keď opustíme režim ZOOM stlačením «SPACE». Môžeme teraz prechádzať celú oblasť 640×400 bodov. Vedľajší príkaz **CLR** a **I** sa vzťahujú na zobrazený výsek.

«L» **LIN**: LINE (kresliť priamku). Ak chceme nakresliť priamku, dáme kurzor do počiatočného bodu želanej priamky a stlačíme FIRE. Potom posunieme kurzor do koncového bodu priamky a znova stlačíme FIRE.

«W» **WIR**: WIRE (kreslenie množiny priamok). Ak bol stlačený FIRE určený pre počiatočný bod, kreslí program vždy jednu

priamku k poslednému koncovému bodu predchádzajúcej priamky.

⟨R⟩ **RAY**: RAY (lúč). Po navolení pevného bodu sa z neho vedú priamky ku kurzoru. Takto dostaneme sieť lúčov.

⟨E⟩ **ELL**: ELIPSA. Ak chceme nakresliť elipsu alebo kružnicu, je potrebné kurzorom určiť jej stred. Potom rozťahujeme bodkovaný štvorec, ktorého strany predstavujú priemery. Môžeme kresliť elipsy do priemeru 255 bodov. V prípade kružnice musia byť obidve dĺžky strán rovnaké. Diely elipsy, ktoré prechádzajú cez rám, sú odrezané (toto platí aj pre iné funkcie).

⟨C⟩ **CUR**: CURVE (krivka). Krivky resp. štvrtelipsy sú potrebné pre zaoblenie rohov.

⟨D⟩ **DSC**: DISC (vyplnená elipsa). Platí ako v príkaze ELL, ale elipsa je vo vnútri vyplnená.

⟨Q⟩ **QAD**: QUADRAT (obdĺžnik). Pre nakreslenie obdĺžnika treba ako pri line fixovať dva body, ktoré predstavujú dva protiležiacie body rohov štvorca.

⟨B⟩ **BLK**: BLOCK (vyplnený obdĺžnik). Podobne ako QAD, ale nakreslí sa vyplnený obdĺžnik.

⟨F⟩ **FIL**: FILL (vyplniť). Táto funkcia vyplní ohraničenú plochu. Plocha sa môže rozprestierať cez všetky štyri dielčie obrazovky, ako je to u väčšiny príkazov. Opätovné stlačenie FIRE túto funkciu ukončí.

⟨P⟩ **PFL**: PATTERN FILL (vyplnenie znakom vzorky). Ohraničenú plochu vyplní program určitým vzorom. Tento je zobrazený v spodnej časti obrazovky. Príkaz platí len pre viditeľnú časť obrazovky. Ak je množina znakov v pamäti príliš veľká, funkcia sa preruší. V takomto prípade treba množinu znakov uchovať na diskete a pamäť vymazať pomocou ⟨Shift K⟩. Pomocou ⟨,⟩ sa grafika vráti do pôvodného stavu.

⟨Shift P⟩ **DPT**: DEFINE PATTERN (definuj plniaci znak). Po každom stlačení FIRE sa kurzor presunie z grafiky do ukazovateľa plniaceho vzoru (16×16 pixelov). Plniace znaky meníme tak, že pri stlačení FIRE pohybujeme joystickom.

⟨*⟩ **SPR**: SPRAY. Funkcia rozosieva pri stlačení FIRE náhodné body okolo kurzora.

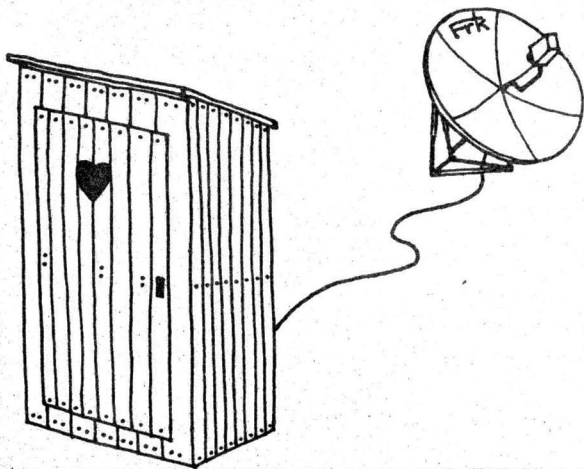
«A» **AND**: AND (logické spojenie AND). Môžeme logicky spojiť (súčin) dve grafické stránky s 320×200 bodmi. Po aktivovaní tohoto príkazu začne blikať výrez, kde sa nachádza kurzor. Pomocou (kurzor vpravo) sa blikajúca časť ďalej spína. RETURN-om sa zvolená obrazovka logicky spojí s práve zobrazenou.

«O» **ODR**: OR (logické spojenie OR). Pracuje podobne ako A.

«X» **XOR**: XOR (logické spojenie EXOR). Pracuje podobne ako A.

«↔» **EXQ**: EXCHANGE GRAFICSCREENS (zámena dvoch grafických stránok). Ak chceme zameniť dve grafické stránky, postupujeme ako pri logických väzbách.

«M» **MOV MOVE** (pohybovať). Funkcia MOVE je najdôležitejšou v celom grafickom programe, lebo tvorí aj základ pre textový režim. Prvým stlačením FIRE deklaruujeme roh výseku obrazovky, ktorou treba pohybovať. Potom rozťahujeme obdĺžnik (maximum 255×255 pixelov), v ktorom je zahrnutý „objekt



presunu". Po zvolení výseku stlačíme FIRE. Pri každom jeho ďalšom stlačení sa výrez obrazovky uloží. Ak je voľba výrezu obrazovky potvrdená chybovým gongom, potom už niet viac miesta v pamäti pre množinu znakov. Vedľajším príkazom «SHIFT» sa zmení v 45° krokoch uhol rotácie, ktorým sa objekt ukladá.

«Z» **TXT**: TEXT. Skôr, ako sa aktivuje textový režim, treba natiahnuť sadu znakov. Po aktivovaní tohoto režimu je kurzor určený pevne a počítač reaguje len na funkcie «CTRL9» «CTRL0» «CTRLK» RETURN DEL a STOP. Aby sa po prípadnom výmaze grafika opäť obnovila, doporučujeme písať v „inverz“ režime“. Z prog-

ramátorsko-technických dôvodov môžeme editovať len riadok, kde sa práve nachádzame. Po stlačení RETURN-u sa začne nový riadok. Rozostup riadkov je určený najvyšším v riadku sa vyskytujúcim znakom.

«Shift Z» **DCH**: DEFINE CHARACTER (definovanie znakov). Tento príkaz zo začiatku pracuje ako MOVE, ale pri výreze grafiky sa mu priradí ASCII kód. Po definícii výseku je potrebné stlačiť tú klávesu alebo jej kombináciu, podľa ktorej sa má grafický výsek objaviť v textovom režime. Vysoký tón gongu potvrdzuje prijatie kombinácie kláves. Povolené kombinácie sú tie, ktoré prijme aj editor BASIC-u (výnimky «CTRL9»=«CTRLR», «CTRL0» RETURN a DEL). Ak sa už po definícii grafického výrezu ozve chybový signál a preruší sa funkcia, to znamená, že pamäť pre znaky je plná. V takomto prípade postupujeme ako pri PATTERN FILL.

«Shift K» **KIL**: KILL CHARACTER MEMORY (výmaz pamäte).

«Shift E» **ECH**: ERASE CHARACTER (výmaz znaku). Vymaže ľubovoľný výrez grafiky z množiny znakov. Ihneď po zadaní príkazu očakáva EGA kombináciu kláves, ktorú treba vymazať.

«Shift U» **UCH**: USE CHARACTER (použitie znaku). Po voľbe používaného znaku možno kurzorom voľne pohybovať ako pri MOVE. Po stlačení FIRE je znak uložený. Týmto príkazom možno simulovať aj štetec.

«Shift M» **MEM**: DISPLAY MEMORY (zobrazenie pamäte). Na obrazovke sa zobrazí voľná časť pamäte znakov. Táto zodpovedá aj MOVE, lebo vo voľnej oblasti sú dočasne uchované aj objekty.

«Shift S» **ZSV**: (uloženie množiny znakov). Po zadaní názvu súboru sa znaky, nachádzajúce sa v pamäti, uložia na disk.

«Shift L» **ZLO**: (načítanie množiny znakov). Hneď po zvolení tohto príkazu sa načíta directory disku. Potom zvolíme súbor pomocou CRSR a definujeme pomocou RETURN-u.

«CBM L» **GSV**: (uloženie grafiky). Po otázke, či má byť uložená obrazovka alebo celá grafika, zadáme názov súboru. Grafika je uložená vo formáte HiEddi (320×200 bodov). Pri voľbe celej grafiky sa uchovávajú štyri jednotlivé obrazovky.

«CBM L» **GLO**: (načítanie grafiky). Načítanie grafiky prebieha ako načítanie znakov. Pri celkovej grafike musíme odpovedať, či sa

má načítať obrazovka jednotlivo, alebo naraz. Jednotlivé časti sa načítavajú vždy do aktuálnej obrazovky.

«**CBM C** **CMD**: **COMMAND** (diskové príkazy). Tento príkaz zasie-
la diskovej jednotke ľubovoľný príkaz DOS.

«**↵** **HSP**: (pomocná pamäť). Grafiku môžeme kopírovať do po-
mocnej pamäte. Ak je v pamäti znakov príliš veľa dát, funkcia
sa preruší chybovým hlášením.

Vedľajšie príkazy

«**CTRL 9**»: zapína reverzný režim pre objekt presunu a text.

«**CTRL 0**»: vypína reverzný režim.

«**CTRL K**»: umožňuje kurzívový výstup objektov a textu. Prvá voľ-
ba spôsobí silnú kurzívu, ďalšie stlačenie zúži kurzívu o polovicu,
tretie stlačenie vráti normálny režim.

«**Shift T**»: otáča objekty a text vždy o 45° v smere hodinových
ručičiek.

«**CLR**»: vymaže aktuálnu stránku grafiky, v režime ZOOM iba vi-
diteľné pole 40×25 pixelov.

«**⌘**»: invertuje viditeľnú obrazovku.

«**S**»: roztiahne obrazovku v smere X alebo Y.

«**T**»: vytvorí zrkadlový obraz grafickej stránky okolo X alebo Y.

«**+**»: zvyšuje rýchlosť behu kurzora.

«**-**»: znižuje rýchlosť behu kurzora.

«**F1**»: zmení farbu kurzora

«**F3**»: zmení farbu popredia

«**F5**»: zmení farbu pozadia

«**F7**»: volí režim znakov: invertovanie, vymazanie a definovanie.
Všetky kresliace funkcie pracujú v aktuálnom režime. Ten je zob-
razený farbou rámu: svetlomodrá – invertovanie, zelená – defi-
novanie, fialová – vymazanie.

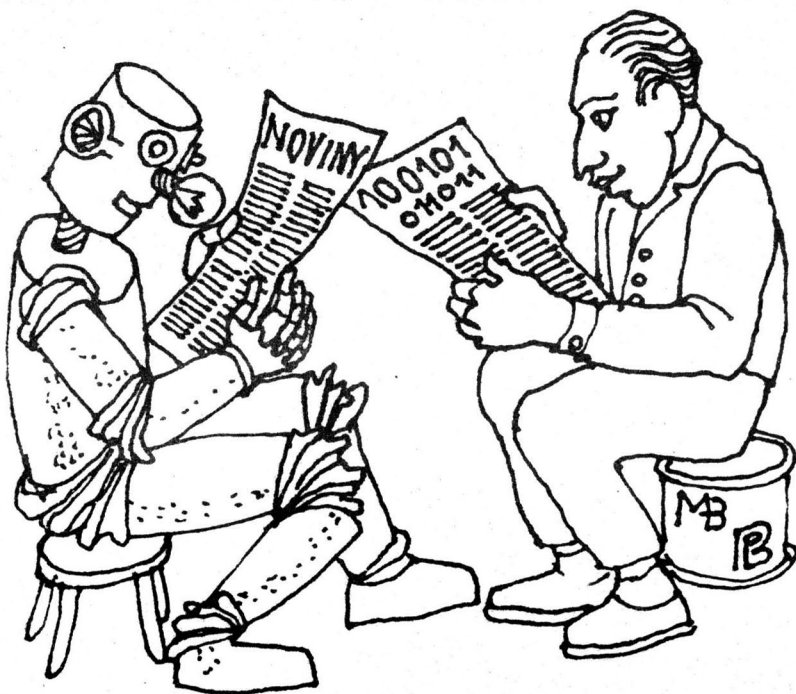
«**Shift 1 až 9**»: zadanému číslu je priradená poloha kurzora.

«**1 až 9**»: kurzor je postavený do definovanej polohy.

«**↵**»: ak je pomocná pamäť obsadená, zamení sa za aktuálnu gra-
fiku. Definovaný obsah dostane pomocná pamäť pomocou «**↵**»,
alebo naplnením vzorku. Ak má byť obsah medzitým prepísaný
načítaním alebo definovaním množiny znakov alebo použitím
funkcie presunu, preruší sa funkcia a zaznie chybový tón.

«**CBM P**»: PRINT. Po zadání tohoto příkazu musíme odpovedať, či chceme vytlačiť jednotlivé obrazovky, alebo celú grafiku. Potom sa načíta z disku rutina EGA – PRINT, a grafika sa vytlačí.

Program EGA V3.2 a jeho dostupné doplnky sú uložené v klubovom archíve.



*V roku 1989 veľa nápadov
a málo porúch*

Vám želá

Commodore klub Košice

Obsah čísel 1–3 Spravodaja

Obvody LSI pre Commodore 64	3
Turbo tape 61k	4
Používanie súborov	6
Úprava magnetofónu pre odposluch	10
Tipy a triky	12
Univerzálny paralelný interface	13
Disková jednotka VC-1541	18
Softverový odposluch Datasetu	21
Tipy a triky	22
Jednoduchý hardverový odposluch	24
Princíp činnosti kompilátorov	24
Využívanie kompilátorov pri tvorbe softveru	26
Sériový interface	30
Conrad Electronic	36
Číselné sústavy	41
Tipy a triky	44
Profi-Ass 64	45
Profi-Mon 64	47
Debugger 64	48
Cofossus 2	49
Grafický editor EGA V3.2	53

Spravodaj COMMODORE č. 3/1988 pre mikroelektroniku a výpočtovú techniku.
Vydáva: Krajský dom pionierov a mládeže – Commodore klub, Maľinovského 2,
Košice.

Vedúci redaktor: Juraj Rusnák. Odborní redaktori: Milan Bobula, Ing. Daniel
Gábor, Ing. Zoltán Rábek, Pavol Žalobín.

Vydávanie povolil VS KNV – odbor kultúry, ev. číslo 25/1988.

Tlač: Východoslovenské tlačiarne z. p., Košice.