

AMIGA



Keď sa rodila naša myšlienka vydávať nový časopis pre šikorý okruh najmä amatérskych užívateľov výpočtovej techniky, uvažovali sme nad koncepciou časopisu, ktorý by sa plne venoval počítačom typu Amiga. Rozhodli však ekonomické otázky, a tak sa Vám dostáva do rúk prvé číslo v takejto podobe, ako ho práve držíte v ruke. Kladieme dôraz na kvalitu príspevkov, k čomu prispela najmä láskavosť nemeckého vydavateľstva Markt&Technik, ktoré nám poskytlo materiál a spoluprácu pri výrobe. Kto už pracoval na počítači Amiga, ten nepochybuje o jeho obrovských možnostiach, výkone a kompatibilitu s celým svetom výpočtovej techniky. Zástupcovia redakčnej rady sa zúčastnili na tlačovej konferencii počas výstavy AMIGA 90 v Kolíne nad Rýnom. Tu sme sa dozvedeli, že firma Commodore predala za posledných 14 mesiacov milión kusov počítačov typu Amiga 500 a trh ani zďaleka nie je ešte naplnený. Amiga 500 sa stala počítačom roku 1989, 1990 a pravdepodobne bude kandidovať na najvyšší post aj v tomto roku. Všetky tieto údaje naznačujú, že orientácia nášho časopisu na tento typ výpočtovej techniky nie je náhodná. Len v Spolkovej republike Nemecko nájdete 5 titulov časopisov venovaných výhradne počítačom Amiga. Za najkvalitnejší z nich sa pokladá časopis "AMIGA" z dielne najväčšieho vydavateľa počítačovej literatúry Markt&Technik. Náš časopis sa Vám bude snažiť poskytovať to najlepšie z najlepšieho.

V našom časopise sa zameriame predovšetkým na mladých adeptov, ktorí ešte len začínajú, a to samozrejme s najlacnejším programovým vybavením Amiga-Basicom, štandardne dodávaným s kúpou počítača. Na stránkach časopisu budeme organizovať kurzy, tipy a triky určené aj pre náročnejších užívateľov, prehľady o najnovších trendoch výpočtovej techniky. Dúfame, že uspokojíme aj profesionálnych užívateľov počítačov typu Amiga. Očakávame, že aspoň trochu utíšime hlad po informáciách z tohto odboru výpočtovej techniky. Radi by sme s pomocou širokej spolupráce čitateľov aktualizovali vybraný okruh problémov. Máme k dispozícii odborníkov, pripravených riešiť Vaše požiadavky, a postupne budeme uverejňovať väčší podiel pôvodnej tvorby od autorov zo Slovenska a z Čiech.

MILAN TUREK

Dejiny jedného úspechu

V júli 1990 oslávil počítač Amiga svoje 5. narodeniny. Kto sa skrýva za koncepciou Amigy? Ako vznikol vysnívaný počítač osemdesiatych rokov? Niekoľko slov o jeho vzniku a jeho ďalšom živote.

Amiga je jedným z najúspešnejších počítačov na trhu domácich počítačov. V Spolkovej republike Nemecko sa predalo vyše 300 000 kusov (Amiga 500, 1.000, 2.000). V týchto dňoch prichádza na trh nový potomok, Amiga 3 000. Víťaznú cestu počítačov Amiga nemožno zabrzdiť. Tohto roku Amiga slávi 5. výročie. Pre náš časopis je to dosť dobrý dôvod na to, aby sme podali dostatočné informácie, ako sa dostal počítač Amiga zrodil.

História Amigy sa začala písať v roku 1982. Traja bohatí Američania chceli investovať peniaze do niečoho nového. Napadlo im, že dajú vyvinúť konzolu pre televíznu hru, ktorá by zatienila všetko, čo na vtedajšom trhu existovalo. A tak títopáni hľadali zopár pracovníkov, schopných projekt uskutočniť. Našli ich v Jaym Minerovi, svojho času vývojovom pracovníkovi firmy Atari, Davidovi Morseovi, ktorý bol manažérom istej firmy na výrobu hračiek a R. J. Micalovi, ktorý do tohto tímu prišiel trochu neskôr ako softwarový návrhár.

Jay Miner celkom nesúhlasil s myšlienkou konzoly pre telehru a tajne vyvinul prístroj, v ktorom bolo naplánované všetko, čo televíznu hru robilo počítačom. V roku 1983 v Amerike klesol záujem o televízne hry, a tak novej firme prišla veľmi vhod Jayho Minerova predvídavosť. Amiga popritom predávala joysticky a hracie moduly cudzích výrobcov, ako aj "joyboard", prístroj, ktorým sa hry mohli riadiť nie ručne, ale nožne. To všetko zabezpečovalo financovanie drahého vývoja počítača.

V januári 1984 na CES (Consumer Electronics Show) v Chicagu predstavili

Amigu vybranému okruhu záujemcov. Počas tohto obdobia pracovali R.J.Mical a Dale Luck vo dne v noci, aby vylepšili software a predviedli počítač na veľtrhu (za hlasnej hudby a postojáčky, aby nezaspali). Záujem odborného publika sa prebudil - obdivovalo najmä grafické schopnosti - ale nenašiel sa nikto, kto by výrobu nového počítača financoval.



Irvin Gould, muž ktorý zabezpečil superpočítač AMIGA pre firmu Commodore

V lete 1984 konečne vyrobili špeciálne čipy počítača Amiga, pričom v januári boli ešte len z platínových a drôtených komponentov. Počítač Amiga znovu predstavili na CES a mnohí odborníci už začali o tomto počítači rozmýšľať, ale opäť sa nenašiel nikto, kto by prevzal financovanie. Pôvodní mecenáši sa financovania vzdali, lebo obchod s počítačom Amiga sa im zdal riskantný. Aby

sa firma udržala nad vodou, zaťažovali spolupracovníci hypotékami svoje domy, zadlžovali sa, no napriek tomu sa situácia vyostrovala a oni sa ocitli pred úplným krachom. Medzičasom sa zakladateľ firmy Commodore Jack Tramiel so svojou firmou rozišiel. Jack Tramiel prisahal firme Commodore pomstu a odkúpil firmu Atari, ktorá stála krátko pred bankrotom, od obra médií Warner Brothers, aby odtiaľto začal obchodný boj proti firme Commodore. Jack Tramiel vedel aj o nastávajúcom bankrote firmy Amiga. Začal vyjednávať o jej prevzatí, pretože potom by mohol veľmi ľahko získať úplne nový počítač. Lenže pri vyjednávaniach vsadil Tramiel na príliš vysokú kartu. Namiesto toho, aby sa pomaly približoval k požiadavkám firmy Amiga, ponúkal pri každom vyjednávaní menej peňazí. Spolupracovníci Amigy boli zo situácie zúfalí, pretože vedeli, že Jack Tramiel bude svoje požiadavky veľmi tvrdo presadzovať, bez ohľadu na koncept firmy Amiga. Záchrana prišla v podobe firmy Commodore, ktorá sa dozvedela o tomto vyjednávaní a zo strachu pred Tramielom ponúkla viac ako dvojnásobok sumy, akú Amiga pôvodne žiadala. Tým

firma Commodore zabila dve muchy jednou ranou. Vyfúkla obchod neobľúbenému bývalému šefovi Tramielovi spred nosa a okrem toho mala vo výhľade nový výkonný počítač, na ktorom už bola urobená veľká časť vývojovej práce. Obchod s C64 síce ešte vždy dobre bežal, ale predsa len bola táto stálica pomaly prekonávaná. Aj firma Commodore zmenila trochu koncept Amigy, čo sa jej vývojovým pracovníkom, ktorí pracovali na projekte s veľkým nadšením, nevelmi

páčilo. Avšak kto platí, ten rozkazuje, a navyše mali aspoň fungujúci hardware, na ktorom mohli stavať. Chýbal im ešte vhodný operačný systém. Každý očakával, že na zimnom CES uvidí nový počítač firmy Commodore, ale predstavenie sa nekonalo. Namiesto neho ukázal Jack Tramiel Atari ST, na ktorom naplno pracovali a ktorý vyvinuli v priebehu troch mesiacov. Commodore zrejme podcenil Atari. Medzičasom vysvitlo, že vývoj multitaskingového operačného systému je ťažší, než sa očakávalo. A ako to už s náhodami býva, anglický Software Haus Metacomco mal v zásuvke operačný systém schopný multitaskingu menom Tripas, ktorého vynikajúcou vlastnosťou bola jednoduchá aplikácia na rôznych počítačoch. Tým dali šancu firme Commodore, aby dostala v relatívne krátkom čase vhodný operačný systém. Vlastné ukladanie Tripasu trvalo len štyri týždne a urobil ho Tim King. King však mohol použiť už existujúce časti operačného systému, a tak bolo vo februári 1985 všetko hotové. S operačným systémom začali určité problémy pre program Amiga, pretože King pracoval s programovacím jazykom BCPL. Ten vykazuje podstatné rozdiely v riadení dát oproti všade používanému jazyku C. Táto malá nedôslednosť systémového programovania sa ťahne operačným systémom až po verziu 1.3, ale od verzie 2.0 mizne, pretože odvtedy je aj "dos. library" písané v jazyku C. 23. júla 1985 konečne nastal deň, kedy počítač Amiga 1 000 ušiel svetlo sveta. Prezentovali ho gigantickou show (okrem iného s Andyom Warholom, ktorý predstavoval grafiku počítača Amiga) v Newyorskej opere tlačí a odbornému publiku. Už v prípravnom štádiu tohto predstavenia kolovalo o novom počítači firmy Commodore veľa fám. Mnohé boli, pravdaže, preháňané a absolútne nezodpovedali pravde. Opona bola zdvihnutá a každý mohol konečne posúdiť skutočné schopnosti počítača Amiga 1 000. Tieto boli skutočne hodné zhladnutia, pretože počítač Amiga bol svojho času o krok vpredu. Na počítač v cenovej skupine okolo 5 000 DM, ponúkal revolučné vlastnosti, ktoré by ste márne hľadali

dokonca aj pri podstatne drahších počítačoch. Obzvlášť vynikajúca bola grafika. Zobrazuje až do 4096 farieb súčasne a maximálne rozlíšenie 600 krát 400 bodov. To bolo na vtedajšie pomery vynikajúce, nehovoriac o neuveriteľnej rýchlosti grafiky, ktorá postavila latku veľmi vysoko vďaka použitiu blittera. Zvuk Amigy spôsobil, že hudba iných počítačov vyznela skôr ako žalostné kvílenie. Štyri digitálne zvukové kanály v stereo-móde umožnili počúvať verný zvuk reči i hudby. K tomu treba ešte pripočítať grafický užívateľský interface, ktorým sa dal počítač pohodlne ovládať. Multitaskin-



Nová AMIGA 3 000 od fy Commodore

gový operačný systém umožňoval, aby sa v pozadí vykonávali vedľajšie úlohy, napríklad tlačenie súborov alebo formátovanie diskiet. Pozoruhodnou zvláštnosťou počítača Amiga je, že procesor nemusí vykonávať prácu sám, ale má na porúdzku zopár mocných pomocníkov - Custom-Chipov.

Súťaž Commodore kontra Atari

Tieto čipy Amige umožňujú pracovať neuveriteľne rýchlo. Môžu súčasne hrať hudbu, robiť čiary a zaplňať plochy,

ako aj nechať rotovať farby bez toho, aby bol zamestnaný procesor 68000. Táto "nezamestnanosť" procesora umožňuje použitie systému s multitaskingom a grafickým užívateľským prostredím. Časovo náročná grafická práca nezaťažuje procesor viac, než je nevyhnutné. Pokiaľ porovnateľné počítače ešte len vypočítavajú priebeh jednej línie, počítač Amiga sa môže venovať dôležitejším úlohám. Okrem toho má Amiga virtuálny koncept displaya. Viacero súčasne prebiehajúcich programov si ťažko môže rozdeliť jedinou obrazovku, ale pri Amige existuje možnosť otvoriť si viaceré obrazovky s rozličnými riešeniami súčasne a súčasne ich zobrazíť na jednom monitore. Od začiatku sa kládol dôraz na to, aby sa operačný software mohol bez ťažkostí rozšíriť nadol kompatibilným spôsobom. Z toho dôvodu sa operačný systém organizoval formou knižníc, zariadení a pravidiel, cez ktoré sa organizujú všetky prístupy k systému. V Amige existuje, na rozdiel od mnohých iných počítačov, len jedna jediná pevná adresa, totiž adresa 4, kde možno nájsť adresu "exec.library". Táto knižnica zodpovedá za systémové riadenie. Pomocou nej získame prístup k ostatným funkciám Amigy. Nejednotnosť vo vývoji systému zapríčňuje známe chybové hlásenie "Guru", ktoré je už vo verzii 2.0 úplne odstránené (pozn. prekl.). Odlišný od "PC škafule" bol aj vývoj skrinky počítača. Dizajnéri navrhli úzku, pekne tvarovanú skrinku, ktorej užitočnosť nemožno podceňovať, pretože aj nepatrným detailom, ako napríklad možnosťou schovať klávesnicu pod počítač, venovali veľkú pozornosť. Všetky dizajnérske návrhy vplyvajú na vysoké hodnotenie Amigy. Operačný systém "studeného štartu" bol ešte dosť zaťažený chybami, a tak ho vydali ako disketovú verziu. Po zapnutí sa síce musel vložiť do pamäti, ale hneď po resete bol opäť k dispozícii. Táto metóda síce nebola pohodlná ako operačný systém v ROM, ale dalo sa tak pohodlne presúpiť na nové, vylepšené verzie. Teoreticky by sa mohli vložiť aj iné operačné systémy, ale nikto nepovažoval za vhodné venovať sa ich vývoju. Mnohí sa zaľúbili do tohto "Porsche"

medzi počítačmi a bolo počuť veľa obdivných vzdychov, keď návštevníci výstavy stáli pred Amigou, ktorá práve predvážala simultánny zvuk a grafiku. Technicky sa dostala do strednej cenovej kategórie. Začiatkom roku 1986 sa prišiel počítač Amiga oficiálne predstaviť do Frankfurtu. Podľa vzoru typickej americkej show inscenovali jeho uvedenie vo frankfurtskej opere. Pre televízne normy PAL, ktoré v Európe prevládajú, boli vyvinuté nové čipy, aby umožnili zobrazenie 512 riadkov. Napriek tomu bolo ešte stále ťažké uviesť Amigu na trh. Na domáci počítač bola príliš drahá-bolo predsa dostať podstatne lacnejší Atari ST. Ten síce neponúkal taký vysoký výkon, ale bol podstatne lepši, ako dovtedy dominujúce 8-bitové počítače. Na profesionálny trh teda Amiga neprenikla, hoci jej technika a výkon vysoko pre-

vyšovali technicky zastaralé PC. Pre Amigu bolo príliš málo profesionálneho programového vybavenia, pretože tu mali jasnú prevahu PC-čka.

Amiga - úspech na celej čiare

Firma vyvinula obrovskú aktivitu, aby urobila Amigu kompatibilnú so štandardom, a zároveň ponúkla pokrokovú techniku a veľkú ponuku softwaru. Na jednej strane existoval software-emulátor "Transformer", ktorý umožňoval spúšťať obmedzený počet PC-programov na počítači Amiga. Avšak toto softwarové riešenie spôsobilo veľké straty rýchlosti, takže na serióznú prácu sa nedalo ani pomyslieť. Na druhej strane navrhli vo firme Commodore v Braunschweigu hardwarový doplnok. "Sidecar" priniesol hardwarovú kompatibilitu PC a Amigy.

Toto riešenie bolo síce výrazne rýchlejšie ako transformer, ale nemohlo Amige pomôcť, aby sa konečne presadila.

Pre veľký útok na trh s počítačmi sa vyvinuli dva nové modely. Amigu 500 Low-Cost, verziu Amigy 1 000, ktorá mala vymeniť v okruhu domácich počítačov C64 a Amigu 2 000, staršieho brata Amigy 500, vyzbrojeného dodatočným "Slotom" a MS-DOS kompatibilitou, ktorý mal preniknúť do oblasti profesionálnych počítačov. Kým Amiga 1 000 bola vybavená len 256 kbyte, vyslala firma Commodore do boja o miesto na trhoch nový počítač Amiga 500 s 512 kbyte a Amiga 2000 s 1 Mbyte RAM do boja o miesto na trhoch. Ako sa nové stroje uchytli, dozvieme sa v niektorom z nasledujúcich čísel.

(T. Podewils, P. Spring, AMIGA 7/90)

ERKA
press

P.O. Box 23
835 32 Bratislava
tel.: (07) 281 414

bankové spojenie:
VÚB Bratislava-mesto
č.ú.: 219648-012

Príjem materiálu: Redakcia preberá príspevky za predpokladu, že ich autorské práva nie sú viazané na tretiu osobu. Súhlas na vytlačenie príspevku považuje redakcia za automatický. Príspevky do redakcie možno poslať aj na magnetických médiách, ktoré sa autorovi vrátia, ostatné príspevky pošleme autorovi len po vyžiadaní, redakcia nezaručuje ich uverejnenie. Po vytlačení príspevku prechádzajú autorské práva výhradne na vydavateľstvo Erka press. Honoráre sú podľa dohody. Ďalšie rozmnožovanie textu a fotografií je možné len s písomným povolením vydavateľstva. Príspevky zasielajte na adresu redakcie.

Cena časopisu: Počítač aktívne vychádza mesačne, cena jedného výtlačku je 18,- Kčs, pri celoročnom predplatnom 17,- Kčs, pri odbere min. 20 ks predplatné 16,- Kčs.

Príjem predplatného: Erka press, P.O.Box 23, 835 32 Bratislava, tel. (07) 28 14 14, fax (07) 375 891. Predplatné posielajte šekom na konto VÚB 21 96 48 - 012 a ústrižok o zaplattení na adresu redakcie.

Inzercia v časopise: redakcia Erka press-EMEL, P.O.Box 23, 835 32 Bratislava. Vybavuje Ľudovít Hamaš, tel. (07) 232 834, fax. (07) 375 891.

Tlač: Západoslvenské tlačiarne Komárno.

Uzavierka čísel: PA máj - 25.3., PA jún - 29.4., PA júl - 20.5., PA august - 24.6., PA september - 22.7., PA október - 26.8., PA november - 23.9., PA december - 21.10.1991.

68040

Generácia nasledujúca po generácii budúcej

Integrovaný obvod MC 68040 zjednocuje funkcie mikroprocesora, jednotky riadenia pamäti a numerického koprocessora. To prináša obrovské zvýšenie výkonu oproti predchodcom rádovo 20 až 40-krát.

Viete, čo znamená pojem CISC? RISC je známejší výraz pre "Reduced Instruction Set Computer" a označuje mikroprocesory, ktoré obsahujú malý počet príkazových viet, ale sú o to rýchlejšie. Pod pojmom CISC nemienia páni "chipdizajneri" nič iné ako tradičné "Complex Instruction Set Computer". V prípade MC 68040 ide o CISC mikroprocesor, ktorý zvládne všetko, a napriek tomu je extrémne rýchly.

Motorola dožičila tomuto čipu vyše milióna tranzistorových funkcií. S materskou doskou ho elektricky spája 179 nožičiek. Jeho výkon je 20 MIPS (Milion Instructions Per Second) a 3,6 MFLOPS (Milion FLOating point Operations Per Second), čo mu zaručuje úctu nejedného RISC-procesora. Kompatibilita s celou rodinou 68- procesorov je ďalšou prednosťou nového super-čipu. Programy, ktoré boli napísané pre MC 68 000 ale bo pre MC 68 020, fungujú - a to oveľa rýchlejšie - aj na MC 68 040. A rýchlosť nie je jediným zlepšením (obr. 1).

Silný trend k vysokej integrácii badať už pri ďalšom vývoji od 68 020 k 68 030, kde jednotka riadenia pamäti "MMU" prešla do procesora. Motorola ide pri 68 040 ešte ďalej: matematický koprocessor (Floating point unit - FPU) nasledoval integráciu MMU. Týmto krokom odpadajú výdavky na hardwarové prepojenie, a tým aj zrýchlenie komunikácie.

Zvýšenie rýchlosti

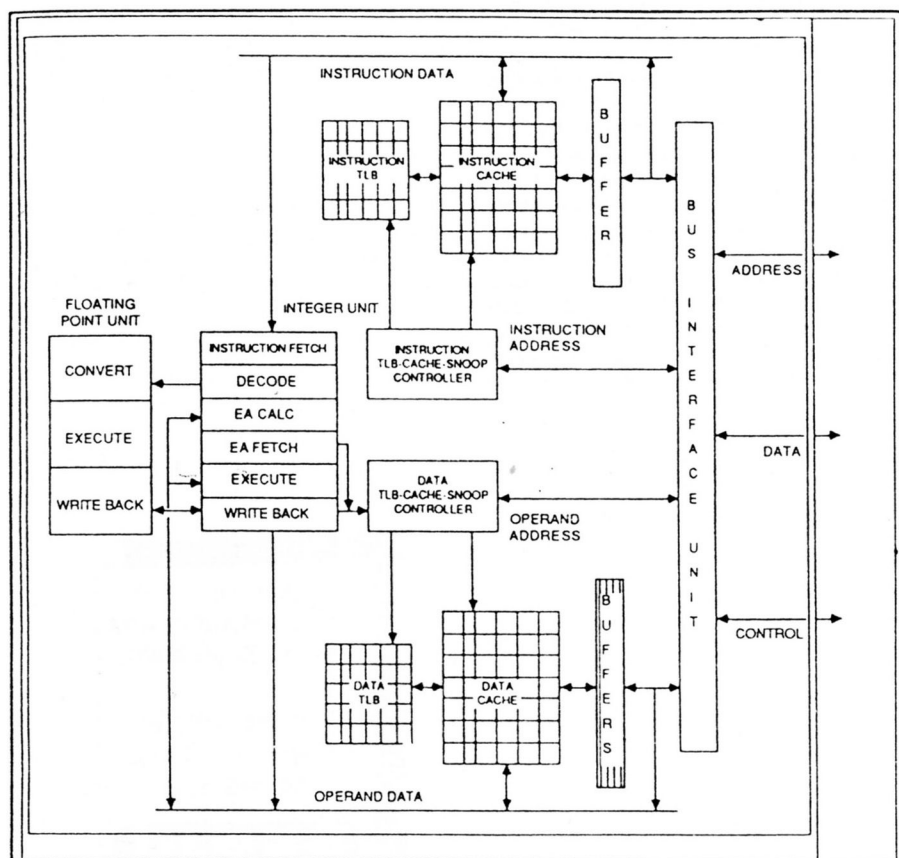
Spolu s integráciou bol prepracovaný aj koncept numerickej jednotky Pipelining. Postup, ktorý už doteraz urýchlil vykonávanie príkazov hlavného procesora, nachádza uplatnenie aj pri FPU. FPU rozdelili do viacerých funkčných blokov, ktoré pracujú paralelne: jeden konvertor, jeden "executor" a jeden obvod Writeback (obr. 2). Spracovanie príkazov sa prekrýva, všetko sa uskutočňuje niekedy až s trojnásobnou rýchlosťou.

Pre záujemcov o technické dáta: FPU má k dispozícii jeden 64x64-bitový hardwarový multiplikátor, jeden 67-bitový Barrel-Shifter pre sériové operácie a jednu 67-bitov širokú aritmetickú

(AU). Všetko má však svoje chybičky krásy. Nie všetky príkazy, ktoré koprocessory MC 68 881 - MC 68 882 ovládajú, je integrovaný FPU schopný vykonať. "Tak predsa len RISC?" Tak trochu áno. Neimplementované príkazy musia byť simulované softwarom.

Redukovaním príkazovej vety trpia najmä transcendentné funkcie. FPU nevie vypočítať sínus ani kosínus. Takže až 2 000-4 000-percentné zvýšenie výkonu sa takto pri programoch Ray-Tracing nerealizuje. Chýbajú logaritmy, odmocniny, ba aj exponenty.

Na čipe našla miesto spolu 8 kbytová pamäť Cache. Jedna polovica slúži ako dátový Cache, druhá ako príkazový Cache. Takýto Cache slúži ako superrýchla dátová pamäť. Keď mikroprocesor vyberá dáta alebo príkazy z pamäti, pripomína si ich pomocou Cache. To enormne šetrí čas, pretože netreba robiť žiadne časovo náročné komunikácie po zbernici. Avšak Cache pamäti prinášajú so sebou niekoľko problémov. Ak sa zmení v pamäti nejaká hodnota, cez Blitter alebo inú jednotku schopnú DMA zapisujú údaje do pamäti.



Obrázok 1: Štruktúra procesora MC 68040 vyzera ako záchranný plán v prípade požiaru

MC 68040 zabudoval pre tieto účely špeciálne zapojenie-hľadač (Snooper). Ak nejaký iný "bus-master" siahne na zbernicu, MC 68040 kontroluje, k akým zmenám došlo. Výrazom "zbernica(bus)" sa označujú elektrické prepojenia, spájajúce dohromady jednotlivé komponenty v počítači. V prípade, že sa zápisné akcie iného "bus-mastera" dotknú miest pamäti, ktoré ležia v Cache, označí ich procesor ako neplatné. Ak siahne na pamäť niekto iný, dotkne sa pritom oblasti Cache, zápis v nej označí za "dirty" a MC 68040 zablokuje pamäť. Potom sám dodá hľadanú hodnotu na bus-master.

Oproti doterajšej realizácii cacheových pamätí ovláda MC 68040 dva rôzne spôsoby pre zásahy do zápisu. Doteraz sa zásah robil zásadne do pamäti a zodpovedajúca hodnota sa aktualizovala v cache. Tento spôsob sa volá pri MC 68040 "writethrough". Druhý spôsob - copyback - zapisuje zmenenú hodnotu len do cache a označí ju tam za "dirty", to znamená, že hodnota v cache nesúhlasí s hodnotou v pamäti. Až neskôr sa prenesie zmenená hodnota do pamäti. To sa

môže stať buď na pokyn CPUSH, alebo vtedy, ak cache pamäť potrebuje priestor pre nové dáta. Pre akcie copyback existuje "Burst-Transfer", pri ktorom sa prenášajú 4 dlhé slová (po 32 bitov) v kuse. "Dirty" data

Už v procesore MC 68030 bolo integrované MMU-Memory managing unit. Je to obvod, ktorého hlavnou úlohou je pripravovať dáta mikroprocesoru. MC 68040 môže adresovať 4 GBytové pamäti. Počítač Amiga s 2 MBytami pamäti môže, ak sa pozerá cez "okuliare" MMU, mať tiež 4 GBytovú pamäť. MMU "prekladá" adresu, ktorú mikroprocesor dodá tak, že ju hodí do pamäti. V skutočnosti možno časť programu umiestniť na pevnom disku - ak ju chce procesor použiť, alarmuje MMU vhodnou rutinou, ktorá vymení potrebný obsah pamäti za terajší. Tak fyzikálna stavba pamäti neobmedzuje veľkosť jedného programu alebo súboru. MMU je okrem toho akýmsi policajtom v počítači alebo skôr strážnou a bezpečnostnou službou. Hermeticky uzatvára oblasti pamäti jedného programu, proti druhým, takže napríklad víry majú malú

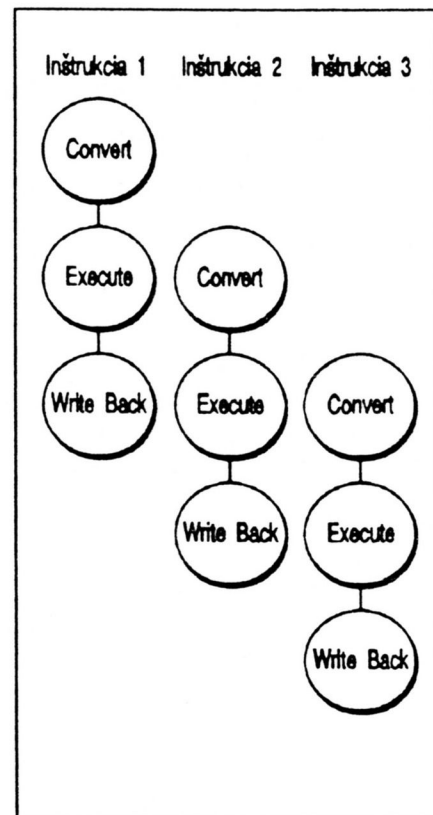
nádej programy infiltrovať. Operačný systém musí samozrejme podporovať MMU, aby mohla byť účinne nasadená. Práve pri počítači Amiga by bola MMU veľkým krokom zabezpečenia proti padaniu systému. Studený štart pri verzii 1.3 však nepodporuje MMU.

Aj vo všeobecnej architektúre sa pri MC 68040 niečo udialo. MC 68030 mal ešte dynamickú zbernicovú štruktúru. To znamená, že pamäť nemusí byť 32 bitov široká, môže mať aj 8- alebo 16-bitovú šírku. Toto slúžilo na zníženie nákladov minimálneho systému. Boot-ROMs alebo Cartridges mohli pozostávať zo 16 bitov širokej pamäte, RAM mohla byť 32 bitov široká. Bolo potrebné, aby sa komunikačný protokol zjednodušil a tým urýchlil vykonávanie operácií. Jeden príkazový cyklus má už len jeden cyklus taktu. To znamená, že 25 MHz procesor môže siahnuť na zbernicu 25 miliónkrát za sekundu. MC 68030 na to potreboval dva cykly, MC 68000 dokonca štyri. To všetko znamená zvýšenie výkonu MC 68040.

Rozšírenú príkazovú vetu a programovanie 68040 opíšeme bližšie v jednom z ďalších vydaní časopisu.

(M. Göckel, AMIGA 4/90)

Obrázok 2: Inštrukcia PIPELINING



Amiga

vo vesmírnych letoch

THREE... TWO... ONE... LIFT-OFF!

Nervozita pri štarte rakety?

Pre organizáciu NASA takýto problém neexistuje.

Štart rakety kontrolujú počítače Amiga.

T"Thirty-one seconds and counting!" Pri tomto ozname hlásať NASA sa každého diváka na myse Canaveral zmocní nepokoj. O pol minúty začne raketoplán alebo raketa neuveriteľnou rýchlosťou stúpať. Len málokto v tej chvíli tuší, aké obrovské náklady sú potrebné na personálne a materiálne zabezpečenie vesmírneho letu. Že sa na tomto scenári podieľajú aj zamestnanci firmy Amiga, vedeli doteraz azda len tí, ktorí s nimi denne v organizácii NASA pracovali.

Od dokončenia programu Apollo začiatkom sedemdesiatych rokov sa americké centrum vesmírnych letov veľmi zmenilo. Dnes existujú dva veľké okruhy: civilná časť organizácie NASA - "Kennedy space center", kde štartujú raketoplány a vojenský okruh riadený veliteľstvom vzdušných síl - "Cape Canaveral". Odtiaľto štartujú všetky nosné rakety s nákladmi úžitkových zariadení pre vojenské, komerčné alebo vedecké účely.

Na území vzdušných síl sa zachovalo niekoľko civilných zariadení a hangárov, z ktorých NASA s firmou McDonnell-Douglas realizuje a sponzoruje spoločnú konštrukciu, testovanie, štart a lety rakety Delta-II. Táto činnosť sa samozrejme nezaobíde bez počítačov. Svoju úlohu zohrávajú aj počítače značky Amiga.

Od marca do mája minulého roku sa na štart s raketou Delta-II pripravoval v izolovanom priestore hangára AE najväčší nemecký výskumný satelit ROSAT (Röntgenový satelit). Nemecký tím

musel úzko spolupracovať s odborníkmi z organizácie NASA, Mc Donnell-Douglas a veliteľstvom vzdušných síl. Tak sa zoznámili s Davem Braunom, ktorý je ako hlavný programátor tvorcom programového vybavenia na počítačoch Amiga, určeného pre organizáciu NASA. Dave prišiel do NASA v roku 1971 ako elektrotechnický inžinier s chabými znalosťami fortranu. Tam ho ihneď zamestnali ako programátora. Mohutné počítače Raytheon 703, neskôr RTS 500 so 16 bitovou zbernicou a pamäťou 128 Kbytov, slúžili do roku 1987. V tomto čase mal Dave už asi jeden rok počítač značky Amiga 1 000. Keď po vyradení starých počítačov Raytheon vo firme uvažovali, čím ich nahradiť, navrhol Dave počítač Amiga. Z počítačov, ktoré prichádzali do úvahy, len Amiga spĺňala dve dôležité požiadavky tvoreného systému "Data Processing", a to schopnosť multitaskingu a detailné informácie od výrobcu o štruktúre zbernice a architektúre podporných obvodov. Najskôr zaobstarali štyri počítače Amiga 1 000, s ktorými mal byť vybudovaný úplne nový systém spracovania dát. Dnes už má systém vyše dvanásť počítačov Amiga 1 000, štyri počítače Amiga 2 000 a štyri externé jednotky Amiga 1 000 vo Vandenbergskom centre vesmírnych letov v Kalifornii a v centre Luis Research v Ohio.

V hangári AE sa zbierajú prúdy dát viacerých systémov, ktoré sa môžu v prípade potreby spracúvať súčasne. Viaceré telemetrické prijímače, ako aj káblové spojenie naplňujú simultánne, v reálnom

čase dáta Atlas-Centaura, Delt-II, úžitkového zaťaženia Delt a raketoplánu do dvojitej paralelnej zbernice (14-bitové adresy, 10-bitové dáta). Zbernica nazývaná "Linkmus", je taktovaná s 250 kHz a zabezpečuje pripojeným počítačom Amiga a DAC (Digitál-Analog-Converter) dáta zo 16 možných vstupov.

Počas odpočítavania "Countdownu" práca pre systém Amiga predstavuje kontrolu, riadenie priebehu a automatické prerušenie odpočítavania;

- prekladanie dát (decimálne, oktálne, percentuálne alebo škálované podľa želania);
- riadenie analógového výstupu (pásový stroje, nekonečný plotter);
- výstup na display;
- pomocné funkcie;
- kontextový help pre užívateľa.

V režime "off-line" sa generujú dátové balíky, ktoré cez modem alebo magnetické pásky dodávajú na analýzu. Ani konverzia do iných formátov (PC IBM, Mac, Tandy, C64) nie je problémom.

Ako teda vyzerá hardware, ktorý toto všetko dokáže? Každý počítač Amiga 1 000 má "pal-box" od firmy "Byte by Byte" s piatimi rozširovacími slotmi podľa štandardu Zorro-I. Tu boli prakticky vyskúšané prvé pal-boxy, a keď sa "Byte by Byte" vzdal výroby, zo zostatkových a náhradných dielov, sa zostavilo ešte šesť jednotiek. V pal-boxoch sa pre každý počítač Amiga nachádzajú dva 180-Mbytové hard-disky od CDC s GVP - kontrolerom (SCSI, podľa štandardu Zorro-II), ako aj jeden 40-Mbytový Ber-

noullibox s výmennými diskami. Najdôležitejšie sú však karty I/O, ktoré si vyvinuli sami (viď ďalej). V počítačoch Amiga podstatne urýchľuje prácu Hurricane-Board so 68020 (14MHz), 68882 (33MHz), 4 Mbyte 32-bitový RAM a 1 Mbyte 16-bitový Fast - RAM. Počítače Amiga 2000 disponujú navyše aj jedným Highspeed-Modemom a jedným AT-Bridgeboardom. Dave a jeho ľudia testujú Beta-verziu pre AT-Bridgeboardu, ktorým sa riadia digitálno-analógové konvertory pre Stripchart-Recorder.

Prvá z troch I/O kariet v počítačoch Amiga, vyvinutá v organizácii NASA, je čistá štvorkanálová vkladacia karta. Dva kanály sú cez vyrovnávaciu pamäť napojené redundantne na obe zbernice. Prístup do počítača je cez Direct Memory Access, preto sa na základnej doske nachádza aj DMA-čip z Motoroly (68450 L10).

Tretí kanál je špeciálny. Je to tzv. "Lift-off" indikátor, čiže signál, ktorý vzniká otvorením kontaktu pri štarte rakety a v počítači má za následok prerušenie, spúšťajúce čas. Tento signál používa počítač ako referenciu počas výstupovej fázy rakety, aby sa na milisekundy presne sledoval plánovaný priebeh štartovacej sekvencie (so zapálením ďalšieho stupňa, odpadnutím vypálených boostrov atď.). Na štvrtom kanále je 10-milisekundové prerušenie, ktoré umožňuje presnú synchronizáciu všetkých pripojených počítačov Amiga.

Najzaujímavejší modul je vybavený rovnakým DMA-obvodom ako základná doska a funguje výlučne ako karta pre výstup. Dva kanály idú paralelne na každú zbernicu, aby spracované dáta podali ďalej na DAC-y. Mnohé dáta PCM sú k dispozícii v 24-bitovom formáte. Pretože kresliace zariadenia sú 10-bitové, posielajú sa z počítača Amiga 3-krát 8-bitov cez menič. V budove a predovšetkým v kontrolných miestnostiach sa nachádzajú tzv. "Display Request Panels", skrátenie DRP. Každý užívateľ, ktorý sedí pri DRP, si môže zvoliť želanú funkciu (napr. generovanie požadovaných dát, hardcopy, formát výpisu). Aktuálnu požiadavku DRP odovzdá počítaču Amiga po údaní čísla prístroja (6-bitová identifikácia) a želanej funkcie (4 bity) ako aj ďalších 22 bitov informácií (číslo strany, špeciálne údaje). Kanál 3 prijíma dopyty až do 32 DRP, kým kanál 4 riadi rovnaký počet video-

generátorov. V počítači Amiga sa v pamäti lokalizujú stránky dát alebo jednotlivé dáta. Videogenerátor, ktorý je priradený k pýtajúcemu DRP, adresuje počítač a oznámi sa mu stránka údajov v pamäti, ako aj počet bytov, aby potom cez štvrtý kanál vyvolal dáta ako text s 21 riadkami po 48 znakov na každej strane.

Každý vytvorený obraz možno na želanie preniesť v kontrolnej miestnosti na veľké obrazovky. 8 televíznych kanálov vedie k 8 km vzdialenému kontrolnému bunkru "Blockhouse" - priamo pri štartovacej veži. Aj tam sa nachádzajú DRP, ktorými možno vyvolať dáta z hangáru AE. Systém nazývaný "CARDS" (Computer Aided Recording and Display System) disponuje vsúčasnosti vyše 2000 preddefinovanými stranami, z ktorých len prvých 500 bolo vypracovaných pre štart nemeckého satelitu s RTG technológiou. Jednotlivé videostrany pre 15 už rozbehnutých alebo rozpracovaných projektov sú uložené na disku a možno ich kedykoľvek vyvolať. Pred štartom má každý z troch aktívnych počítačov Amiga cca 3,5 Mbyte vo forme systémových dát a televíznych strán v pamäti, takže ich možno vyvolať v reálnom čase.

Tretia karta v počítači Amiga 1 000 ponúka päť sériových duplexkanálov. Tri z nich zásobujú vyše Syncmodemy s výkonom vyše 56 kbit/s externé počítače vo Vandenbergu a Clevelande. Dráhu údajov 56-kbit/s nevybrali z technických dôvodov, ale preto, že bola najlacnejšia. Dva zvyšné kanály slúžia na rýchlu komunikáciu medzi prioritnými počítačmi Amiga (tri operačné systémy nasadené počas štartu) s prenosovou rýchlosťou 10 Mbit/s.

Z dvanástich počítačov Amiga 1 000, ktoré sú k dispozícii, sa teda používajú tri na operácie, tri na programovanie a jeden na vývoj hardwaru. Ostatné slúžia na inicializovanie systému, na analýzu dát alebo ako rezerva.

Každý počítač sa dá napojiť na viaceré systemizéry cez sériové alebo paralelné rozhranie. Sú to malé prístroje, obsluhujúce všetky pomalé periférne zariadenia, napr. všetky tlačiarne a "Launch Vehicle Progress" - údaje v kontrolnej miestnosti. Je to display, na ktorom sú znázornené všetky dôležité časové údaje (Countdown - hodiny, čas dohorenia jednotlivých stupňov, koordináty dráh atď.). Každý systemizér ob-

sahuje 1 Mbyte Printer buffer, ako aj 256 kbyte displaybuffer.

Aby pri náhlom výpadku prúdu nedošlo k zrúteniu systému, každý počítač je napojený na sieť neprerušiteľnej dodávky energie (UPS).

Amiga na obežnej dráhe

Pri výpadku energie UPS prevezme zásobovanie elektrickou energiou v priebehu 1/60 sekundy a počítač Amiga môže udržať pri živote najmenej 10 a maximálne 30 minút.

Dave Braun napísal za vyše dva roky okolo 80 % softwaru. Ďalších dvoch spolupracovníkov dostal až viani v januári. Vývojoví pracovníci a nákupcovia Jerry Michaelle ako aj Hal Greenlee boli pritom hneď od začiatku. Dave a jeho mužstvo sú hrdí, že pri programovaní s výnimkou programov pre aritmetický koprocesor (Floating-point výpočty) bol použitý len systém knižníc, takže aktualizovanie a obnova modelu (Amiga 3 000) by nemal byť veľký problém. Programuje sa v jazyku C, toho času Manx Aztec C 50 a s editorom Cygnus. Kvôli kratšiemu kódu dali prednosť Manxu pred Lattice-C Compilerom. Programovanie v asembleri zavrholi ako príliš obtiažne a neprehľadné. Rozhodujúce boli aj zlé skúsenosti s predchádzajúcou generáciou počítačov Raytheon, ktorého programy v asembleri nemohli byť prebraté. Ešte koncom minulého leta im dodali tri počítače Amiga 3 000, ktoré do konca roka rozšírili o "Bernoulliboxy", doplnené optickými diskami s možnosťou zápisu a čítania.

Pretože sa čoraz viac pracovných skupín rôznych raketových systémov (napr. Titan) zaujíma o prenos dát v hangári AE - tam sa čoraz dôležitejšiu stáva kontrolný systém GPS (Global Positioning System) - nebude asi dlho trvať, kým príde čas využitia počítačov Amiga aj v iných systémoch. Nasadenie počítačov Amiga v takej dôležitej a kritickej oblasti, akou sú lety do vesmíru, poukazuje na schopnosti tohto počítača.

Počítač Amiga však nenájdete len vo vesmírnych projektoch. Ak by ste prišli na Floridu ako turista, pravdepodobne navštívite aj filmové štúdiá MGM Walta-Disneyho v Orlande. Ak si potom pozriete Indiana-Jones- Stuntschov, všimnite si okraj javiska. Ktoré počítače asi tak riadia toto náročné predstavenie?

(K. Steinberg, AMIGA 8/90)

Checksummer

CHECKIE 42 ver. 1.1

Tento program je neoceniteľným pomocníkom pri zadávaní listingov do počítača. Pomáha odstraňovať chyby pri preklepoch a šetrí mnoho cenného času.

Kto skúšal opísať dlhší listing programu bez chyby, už vie, že je to prakticky nemožné. Prinášame program CHECKIE 42 vo vynovenej verzii 1.1, ktorý používajte pri prepisovaní našich listingov. Ušetríte si veľa času pri nekonečnom hľadaní chyby preklepu a aj nervov.

Forma listingov

Riadok listingu pozostáva z jednej, najviac štvormiestnej číslice, čísla riadku, dvoj- prípadne trojznakovkej kontrolnej sumy a samotného riadku programu.

Príklad:

10 TIO print "Hallo!"

Po prázdnej pozícii za číslom riadku sa nachádzajú 3 znaky kontrolného kódu. Jednotlivým znakom môže byť číslica (0-9), malé písmeno (a-z) alebo veľké písmeno (A-Z). Prvé dva znaky predstavujú samotný kontrolný kód. V treťom znaku je uložená pozícia stĺpca prvého neprázdneho znaku. Je to zaujímavé najmä pre užívateľa zvyknutého na členenie a ľahšiu identifikáciu programových cyklov. Ak nechcete odlišovať cykly v priebehu programu, po zadaní prvých dvoch znakov kontrolovanej sumy môžete, zadávanie ukončiť s tlakom klávesu "Return".

Narábanie s programom CHECKIE 42

Po štarte chce od Vás program meno dátového súboru. Zadané meno používa CHECKIE 42 k zapamätaniu riadkov listingu. Ak už existuje program s uvedeným menom, CHECKIE poskytuje

po otázke "Iba zadanie kontrolnej sumy?" dve možnosti:

-a- zadanie dát s kontrolnou sumou na obrazovku alebo na tlačiareň

-n- načítanie programových riadkov z dátového súboru, ktorý je poruke a zadanie kontrolnej sumy pomocou klávesnice.

Obidve alternatívy sú určené pre užívateľov, ktorí nechcú používať riadkový editor CHECKIE, ale rýchlejší a pohodlnejší editor iného typu, napr. z AmigaBasicu /zapamätanie "...", a/.

Po stlačení -a- vypočíta CHECKIE kontrolnú sumu Vášho textu, ktorú si môžete porovnať s našim listingom. Pri zadaní výpisu na obrazovku, CHECKIE zapisuje dáta súčasne aj na disketu s koncovkou .chk. Tieto dáta sa dajú jednoducho prezrieť v CLI príkazom TYPE.

Ak ste na otázku "Zadanie z dát" odpovedali -a-, potom Vám stačí už len zadávať kontrolnú sumu. Ak zadané kontrolné číslo nezodpovedá vypočítanému, môže sa riadok korigovať priamo.

Ak odpoviete na obidve otázky -n-, CHECKIE čaká na konci udaného programu na číslo riadku -celkový počet + 1- na zadávanie nového riadku. Všetky zadávané dáta pripojí CHECKIE k danému programu. Táto funkcia je veľmi užitočná, keď máte listing rozdelený na viacero častí.

Ako vkladáme listing

Ak ste na začiatku zadali dátové meno, alebo ak pridávate k určitému programu nové riadky, tak pracujete v normálnom zadávacom režime. CHECKIE 42 automaticky ponúka čísla riadku a čaká na zadanie kontrolnej sumy. Až po stlačení klávesnice "Return" počíta CHECKIE kontrolnú sumu. Prázdne miesta pred a za príkazovým riadkom ignoruje. Ak súhlasí zadaný programový riadok s kontrolnou sumou, ktorá je v listingu, zapamätá si CHECKIE zadanie a čaká na ďalší programový riadok.

Modus pripojenia:

Môže sa stať, že pri zadávaní riadku dôjde k chybe. CHECKIE 42 vráti kurzor na začiatok riadku a čaká na korekciu. Zmeny sa dajú jednoducho korigovať pomocou klávesou "Backspace" alebo "Delete". Stlačením funkčného klávesu "F2" zapneme modus pripojenia, ktorý však po korekcii chyby musí byť opäť vypojený, pretože spomaľuje zadávanie. V prípade, že chcete urobiť svoju vlastnú zmenu programového riadku, napr. napísaním vlastného komentára, CHECKIE to umožní ignorovaním kontrolnej sumy stlačením funkčného klávesu "F6". Túto funkciu však odporúčame používať veľmi opatrne, aby sa nestalo, že CHECKIE prevezme falošnú kontrolnú sumu. Po skončení zadávania programových riadkov Vám CHECKIE oznámi, koľko nekontrolovaných riadkov prevzal. Prirodzene, môže vzniknúť situácia, keď programový riadok je správne napísaný, ale do kontrolovanej sumy sa nám dostala chyba. Po stlačení funkčného klávesu "F1" môžeme kontrolnú sumu korigovať. Funkčný kláves "F7" slúži na zmenu odporúčaných očíslovaní riadku. Takto môžeme preberať len určité časti listingu v programe, kde potrebujeme kontrolovať len niekoľko programových úsekov.

Ukončenie zadávania:

Ukončenie zadávania vykonáme kombináciou klávesou "Ctrl e". Kto už niekedy hľadal chybu preklepu v trošku dlhšom listingu, nám potvrdí, aká je to nepríjemná práca. Preto odporúčame zadávať dlhšie programy pomocou kontrolných súm prostredníctvom programového vybavenia CHECKIE 42.

(Program je publikovaný na ďalších stranách.)

```

REM *****
REM ***      Checksummer: Checkie 42      ***
REM ***      Version 1.1      sl.      ***
REM *****
Start:
  GOSUB Init
  GOSUB OtvorData
  IF dn$="" THEN Koniec
  GOSUB Obraz
NovyRiadok:
  GOSUB mazat
  GOSUB ZadanieSummy
  IF FKoniec=true THEN Koniec
Opakovanie:
  GOSUB ZadanieRiadku
  IF FKoniec=true THEN Koniec
  GOSUB CalcSumme
  IF FZok = falsch THEN GOTO Opakovanie
  GOSUB Prebratie
  GOTO NovyRiadok
Koniec:
  GOSUB hotovo
  END
Init:
  true=-1
  false=0
  Lriadok=240
  LBriadok=60
  PocBriadky=Lriadok/LBriadok
  zx=6 : zy=14
  sx=20 : sy=5
  PocCsZ=3
  DIM z(Lriadok)
  DIM cs(PocCsZ*2)
  cs(PocCsZ)=0
  a=0 : b=0 : c=0
  i=0 : j=0 : k=0
  FZok = true : FCz3=100
  Checkfile=0 : riadok=1
  READ Faktor(i)
  WHILE Faktor(i)<>0
    i=i+1
    READ Faktor(i)
  WEND
  PocFak=i
  DATA 2,3,4,5,6,0
  RETURN
OtvorData:
  ON ERROR GOTO ChybaDat
  CLS
  LOCATE 3,5:dn$="":INPUT "Meno suboru: ";dn$
  OPEN dn$ FOR INPUT AS #1
  IF FChyba=53 THEN NoveData
  LOCATE 5,5
  PRINT "Iba zadanie kontrolnej sumy? (a/n) "
  e$=INPUT$(1)
  IF e$="a" THEN
    LOCATE 7,5:e$=""
    PRINT "Obrazovka alebo tlaciaren? (o/t) "
    e$=INPUT$(1)
    IF e$="t" THEN
      OPEN "prt:" FOR OUTPUT AS #4
    ELSE
      OPEN dn$+".chk" FOR OUTPUT AS #4
    END IF
  WHILE NOT EOF(1)
    LINE INPUT#1,e$
    apos=LEN(e$)
    z(apos)=32
    FOR i=1 TO apos
      z(i-1)=ASC(MID$(e$,i,1))
    NEXT i
    GOSUB CalcSumme
    i=0
    WHILE z(i)=32 : i=i+1 : WEND
    IF FCz3=i THEN

```

```

cs(6)=-16
ELSE
  cs(6)=i : FCz3=i
END IF
PRINT USING "####";riadok; : PRINT " ";
PRINT#4,USING "####";riadok; : PRINT#4," ";
FOR i=4 TO 6
  IF cs(i)>35 THEN cs(i)=cs(i)+6
  IF cs(i)>9 THEN cs(i)=cs(i)+7
  PRINT CHR$(cs(i)+48);
  PRINT#4,CHR$(cs(i)+48);
NEXT i
PRINT " ";e$
PRINT#4," ";e$
riadok=riadok+1
WEND
CLOSE 1 : CLOSE 4
dn$=""
PRINT:PRINT "Stlac lub.klavesu"
e$=INPUT$(1)
RETURN
END IF
LOCATE 7,5
PRINT"Zadanie z datoveho suboru? (a/n) "
e$=INPUT$(1)
IF e$="j" THEN
  Checkfile=1
  CLOSE 1
  GOSUB backup
  OPEN dn$+".bak" FOR INPUT AS #1
  OPEN dn$ FOR OUTPUT AS #2
  RETURN
END IF
WHILE NOT EOF(1)
  LINE INPUT#1,e$
  riadok=riadok+1
WEND
CLOSE 1
NoveData:
OPEN dn$ FOR APPEND AS #2
RETURN
ChybaDat:
FChyba=ERR
IF FChyba=53 THEN RESUME NEXT
ON ERROR GOTO 0
backup:
OPEN dn$ FOR INPUT AS#1
OPEN dn$+".bak" FOR OUTPUT AS #2
flen=LOF(1)
WHILE flen>=32000
  e$=INPUT$(32000,1)
  PRINT#2,e$;
WEND
IF flen THEN
  e$=INPUT$(flen,1)
  PRINT#2,e$;
END IF
CLOSE 2 : CLOSE 1
RETURN
Obraz:
CLS
LOCATE 2,10
PRINT ">>>>>>>" Checkie 42";
PRINT " - Vypocet kontrolnej sumy! <<<<<<<"
LOCATE 13,zx
FOR i=1 TO 2
  FOR j=1 TO LBriadok
    PRINT "-.";
  NEXT j
  LOCATE 13+PocBriadyky+1,zx
NEXT i
LOCATE 5,30:PRINT "<CTRL e> = Ukoncenie programu"
LOCATE 7,8:PRINT "<F2> = Mod plnenia"
LOCATG 7,40:PRINT "<F1> = Zmena kontrolnej sumy"
LOCATE 9,40:PRINT "<F6> = Zapamatanie riadku"
LOCATE 9,8:PRINT "<F3> = Vymazanie riadku"

```



```

LOCATE 11,8:PRINT "<F7> = Zmena cisla riadku"
RETURN
mazat:
LOCATE zy,1
FOR i=1 TO PocBriadky
  PRINT TAB(zx);SPACE$(LBriadok)
NEXT i
FOR i=0 TO Lriadok
  z(i)=32
NEXT i
LOCATE zy,1
PRINT USING "#####";riadok
apos=0
RETURN
ZadanieSumy:
Farbe=0 : i=1
LOCATE sy,sx+1
PRINT "...
Blikanie:
Farbe=Farbe XOR 1
COLOR Farbe,0
LOCATE sy,sx-15
PRINT "Kontrolna suma:"
LOCATE sy,sx+i
e$=INKEY$
IF e$="" THEN Blikanie
COLOR 1,0
e=ASC(e$)
IF e=5 THEN
  FKonic=true : REM <CTRL e>
  i=PocCsZ+1 : REM hotovo
ELSEIF e=135 THEN
  GOSUB Novyriadok
  e=135
ELSEIF e=8 THEN
  IF i>1 THEN
    i=i-1 : REM <BS>
    LOCATE sy,sx+i : PRINT "."
  END IF
ELSEIF e=13 THEN
  IF i=PocCsZ THEN i=PocCsZ+1 : REM <CR>
ELSE
  IF e>47 AND e<58 THEN
    e=e-48 : REM 0-9
  ELSEIF e>64 AND e<91 THEN
    e=e-55 : REM A-Z
  ELSEIF e>96 AND e<123 THEN
    e=e-61 : REM a-z
  ELSE
    GOTO Blikanie
  END IF
  PRINT e$;
  cs(i)=e
  i=i+1
END IF
IF i<=PocCsZ THEN Blikanie
ESKonic:
COLOR 1,0
LOCATE sy,sx-15
PRINT "Kontrolna suma:"
RETURN
Novyriadok:
IF FZok = true THEN
  Novyriadok=0
  WHILE e<>13 OR Novyriadok=0
    LOCATE zy,1:PRINT USING "#####";Novyriadok;
    e=ASC(INPUT$(1))
    IF e>47 AND e<58 THEN Novyriadok=Novyriadok*10+e-48
    IF Novyriadok > 9999 THEN e=8
    IF e=8 THEN Novyriadok=INT(Novyriadok/10)
  WEND
  IF Checkfile THEN
    IF Novyriadok < riadok THEN
      WHILE NOT EOF(1)
        LINE INPUT#1,e$
        PRINT#2,e$

```

```

WEND
CLOSE 1 : CLOSE 2
GOSUB backup
OPEN dn$+".bak" FOR INPUT AS #1
OPEN dn$ FOR OUTPUT AS #2
riadok=1
END IF
WHILE (Novyriadok > riadok) AND (NOT EOF(1))
  LINE INPUT#1,e$
  PRINT#2,e$
  riadok=riadok+1
WEND
IF EOF(1) THEN
  CLOSE 1
  Novyriadok=riadok
  LOCATE zy,1:PRINT USING "#####";Novyriadok;
  Checkfile=0
END IF
END IF
riadok=Novyriadok
END IF
RETURN
ZadanieRiadku:
x=cs(PocCsZ)
Dalej:
cy=zy+INT(x/LBriadok):cx=zx+(x MOD LBriadok)
LOCATE cy,cx
COLOR 0,1
PRINT CHR$(z(x));
LOCATE cy,cx
IF x>apos THEN apos=x
IF Checkfile AND FZok THEN
  IF EOF(1) THEN
    Checkfile=0 : CLOSE 1
  ELSE
    e$=INPUT$(1,1)
  END IF
ELSE
  e$=INKEY$
END IF
IF e$="" THEN Dalej
COLOR 1,0
PRINT CHR$(z(x));
LOCATE cy,cx
e=ASC(e$)
IF ((e AND 127)<32) OR e=127 THEN Controlcode
IF imode THEN GOSUB insert
PRINT e$
z(x)=e : e=30
Controlcode:
IF e=13 OR e=10 THEN
  RETURN
ELSEIF e=30 THEN
  a=1
ELSEIF e=29 THEN
  a=LBriadok
ELSEIF e=31 THEN
  a=-1
ELSEIF e=28 THEN
  a=-LBriadok
ELSE
  GOTO noCrs
END IF
x=x+a
IF x>=0 AND x<Lriadok THEN Dalej
x=x-a
GOTO Dalej
noCrs:
IF e=8 THEN
  IF x>0 THEN
    x=x-1
    LOCATE zy+INT(x/LBriadok),zx+(x MOD LBriadok)
    FOR i=x TO apos
      z(i)=z(i+1)
      PRINT CHR$(z(i));
      IF i MOD LBriadok=59 THEN PRINT:PRINT TAB(zx);

```

```

NEXT i
z(apos)=32 : PRINT " "
apos=apos-1
END IF
ELSEIF e=127 THEN
FOR i=x TO apos
z(i)=z(i+1)
PRINT CHR$(z(i));
IF i MOD LBriadok=59 THEN PRINT:PRINT TAB(zx);
NEXT i
z(apos)=32 : PRINT " "
apos=apos-1
ELSEIF e=129 THEN
GOSUB ZadanieSumy
x=cs(PocCsZ)
ELSEIF e=130 THEN
imode=imode XOR 1
LOCATE 7,28
IF imode THEN
PRINT "von"
ELSE
PRINT "dnu"
END IF
ELSEIF e=131 THEN
GOSUB mazat
x=cs(PocCsZ)
ELSEIF e=134 THEN
RETURN
ELSEIF e=5 THEN
FKoniec=true
RETURN
END IF
GOTO Dalej
insert:
IF apos>x THEN
FOR i=apos TO x STEP -1
z(i+1)=z(i)
NEXT i
z(x)=32
apos=apos+1
IF apos=Lriadok THEN apos=apos-1:z(Lriadok)=32
FOR i=x TO apos
PRINT CHR$(z(i));
IF i MOD LBriadok=59 THEN PRINT:PRINT TAB(zx);
NEXT i
LOCATE zy+INT(x/LBriadok),zx+(x MOD LBriadok)
END IF
RETURN
CalcSumme:
a=0 : b=0 : c=0
IF e=134 THEN
FZok=true
FF6=FF6+1
ELSE
WHILE z(apos)=32 AND apos>0
apos=apos-1
WEND
IF apos>0 THEN
WHILE z(c)=32
c=c+1
WEND
END IF
FOR i=c TO apos
j=(i-c) MOD PocFak
k=(i+1-c) MOD PocFak
a=a+((z(i) AND 127)-32)*Faktor(j)
b=b+((z(i) AND 127)-32)*Faktor(k)
NEXT i
cs(4)=a+riadok-INT((a+riadok)/62)*62
cs(5)=b+riadok-INT((b+riadok)/62)*62
FZok=(cs(1)=cs(4)) AND (cs(2)=cs(5))
END IF
RETURN
Prebratie:
FOR i=0 TO apos
PRINT#2,CHR$(z(i));

```

```

NEXT i
PRINT#2,""
riadok=riadok+1
RETURN
hotovo:
IF Checkfile THEN
WHILE NOT EOF(1)
LINE INPUT#1,e$
PRINT#2,e$
WEND
CLOSE 1
END IF
CLOSE 2
CLS
LOCATE 12,35
PRINT "H O T O V O !!!"
LOCATE 20,1
IF FF6<>0 THEN
PRINT "P o z o r !!! ";
PRINT FF6;" Riadky budu zapamatane bez kontroly!"
END IF
RETURN

```

Tipy a triky

Pohyb obrazovky

Všetko je záležitost intuície. Uvádžam príklad, čo všetko možno spraviť cez rutinu "intution". Skúsme teda s myšou jednoducho pohybovať celou obrazovkou.

```
DECLARE FUNCTION ViewAddress& LIBRARY
LIBRARY intuition.library
vi&=ViewAddress&
WHILE MOUSE(0)=0
  dx%=66+MOUSE(1)/5.8
  dy%=14+MOUSE(2)/2.5
  IF dx%>128 THEN dx%=128
  POKEW vi&+12,dy%
  POKEW vi&+14,dx%
  CALL remakedisplay
WEND
LIBRARY CLOSE
END
```

Tento program umožňuje vycentrovať obraz na monitore pomocou myši. Tento príklad Vám určite poslúži ako základ pre ďalšie experimentovanie s funkciou

Remake Display

Tento trik sa zaoberá zmazaním alebo kopírovaním väčšieho úseku basicovského programu. Označme potrebný dátový blok bez toho, aby sme museli neustále držať stlačené ľavé tlačidko na myši. Zайдite najprv ako zvyčajne s myšou na želaný úsek programu, ktorý chceme označiť. Ako obvykle zatlačíme ľavé - pozor nepustiť - a potom pravé tlačidko na myši. Stlačili ste obidve tlačidlá? Pustite ľavé pred pravým. A teraz môžete pohodlne s myšou označiť želaný úsek dát alebo riadkov programu. Obidve tlačidlá sú voľné. Na konci označeného bloku si zopakujte hru, ibaže v opačnom poradí:

- pravé tlačidko na myši držať stlačené
- takisto stlačiť ľavé
- najprv pustiť pravé.

Teraz je želaný blok označený. Ale pozor, počas celej procedúry nesmiete stlačiť žiadny iný kláves.

Rýchlejšie listovanie v editore Basicu

Klávesnice - Kurzor hore - Kurzor dole - v spojení so "Shiftom" slúžia k listovaniu strán basicovského editora. Keď máme dlhší program, cca 60 strán alebo aj viac, trvá dosť dlho, kým si nalistujeme žadanú stranu. Väčšiu rýchlosť listovania možno dosiahnuť, ak stlačíme súčasne kláves "Shift" a "Kurzor"

vpravo - 2x a potom listujeme obvyklým spôsobom, ale podstatne rýchlejšie na prázdnej obrazovke. Naspäť sa vrátíme cez kláves "Shift" a dvojitém stlačením klávesu "Kurzor vľavo".

"Čary máry fuk ..." a myš je preč

Príklad malého gagu pre programátorov v Basicu:

```
LIBRARY "graphics.library"
CALL FreeSprite /O/
```

Program nám odčaruje šípku (myš) z obrazovky. Ale nie sú to nijaké čary. Len čo sa myši opäť dotknete, šípka sa na obrazovke zjaví.

Disketi

Pravdepodobne každý programátor zažíva denne stresy, keď hľadá určitý program a nevie na ktorej diskete sa nachádza. Nezostáva mu nič iné, len práčne prezerať "podozrivý" disk za druhým na obrazovke počítača. Program DISKETI, ktorý ponúkame vo vylepšenej a preloženej verzii, pomáha pri archivovaní diskiet a výpisom obsahu adresára na etiketu diskety ponúka ľahký a rýchly prehľad o obsahu diskety.

Narábanie s programom

Program je vyhotovený v AmigaBasicu, kde po spustení vypíše na obrazovke úvodné informácie. Obsluhuje sa veľmi jednoducho cez pomocné okná. Tu sa dá meniť aktuálny adresár, môžeme meniť meno programu, program umožňuje prácu s externým diskovým zariadením. Výpis etikety umožňuje prácu max. 32 titulov programov, pričom nevypisuje názvy jednotlivých knižníc. V najhornejšom okienku poskytuje informácie o tom, čo práve robí, takže keď prebieha triedenie väčšieho počtu programov, treba byť trpezlivý, lebo to zaberá viac času. Programy sú zoradené v abecednom poradí. Ak DISKETI ukončil triedenie obsahu diskety, ponúka pomocné okienko s otázkou na výtláč obsahu adresára.

Tu je možnosť nastavenia on-line režimu tlačiarne, ak sme to ešte nestihli urobiť. Veľkosť typu písma výpisu obsahu direktory sa automaticky vyberá podľa množstva editovaných programov. Výpis je najideálnejšie robiť na samolepiace etikety, ktoré dostať na páse po 150 ks. Potom stačí len odlepiť vypísanú etiketu z nevoskovaného papiera a nalepiť ju na požadovanú disketu.

Listing programu odporúčame prepísať do počítača pomocou programu Checkie 42. Program potrebuje k svojej činnosti knižnicu dos.bmap a exec.bmap.


```

1 Xo0 *****
2 R6 *****Program na vytlac Disketikety *****
3 Zq *****
4 ee 'modifikovana verzia.....8.5.89
5 Jq '
6 Bb CLEAR ,30000
7 jL DEFNG a-z
8 Fn PALETTE 0,0,0,.7
9 oG PALETTE 1,.9,.9,.9
10 ty PALETTE 2,0,0,0
11 MZ LIBRARY "intuition.library"
12 6C LIBRARY "graphics.library"
13 Kz WINDOW CLOSE 1
14 dW WINDOW 2,,,0
15 Gt okno&=WINDOW(7)
16 iB POKEW okno&+98,0
17 uj CALL RefreshWindowFrame&(okno&)
18 lQ Sprava:
19 fd2 LOCATE 3,15
20 cO Tien "Srdecne vas vita program - DiskEti!",0
21 hK PRINT
22 vD Tien "Najprv zopar malych.poznamok:",0
23 Vv Tien "Nacitavam direktory vopred ",1
24 vG Tien "urcenej diskety, pritom",0
25 Hl Tien "si treba uvedomit nasledujuce:",0
26 mP PRINT
27 cT Tien "Nacitavam maximalne 200 suborov pre 1
directory.",0
28 Jv Tien "Vo vypise neukazujem ziadne meno
directory.",0
29 KB Tien "Neukazujem ziadne .info data vo
vypise.",0
30 rV Tien "Mena su dlhe maximalne 15 znakov.",0
31 rU PRINT
32 HV Tien "Data su abecedne zoradene ",1
33 qX Tien "a jednotlivy vypisane. Potom mozete",0
34 yM Tien "tieto data editovat, preskakovat, ",1
35 nO Tien "davaj do vytlace",0
36 G7 Tien "alebo priamo menit v nasledujucej ",1
37 fy Tien "subdirektory. Zmena",0
38 Uu Tien "sa deje automaticky na konci kazdej
direktory. ",1
39 BK Tien "Vami zadane data",0
40 49 Tien "budu trvale ukazovane!",0
41 8b Tien "Meno diskety moze byt takisto
vytlacene!",0
42 Aa Tien "(toto mozete ale zmenit!)",0
43 3g PRINT
44 ve Tien "POZOR!!! Beriem maximalne 32
datovych mien",0
45 Cp Tien " ...stlacte lave tlacitko na
mysi!",1
46 rM Mys
47 JP CLS
48 7P LOCATE 12,29
49 J9 Tien "... este chvilu strpenia!",0
50 EW LOCATE 14,18
51 8H Tien "musim najprv nacitat .bmap
programy...",0
52 lO LIBRARY "dos.library"
53 rU LIBRARY "exec.library"
54 jV DECLARE FUNCTION ExNext% LIBRARY
55 tR DECLARE FUNCTION Lock LIBRARY
56 Ad DECLARE FUNCTION AllocMem LIBRARY
57 fm DECLARE FUNCTION IoErr% LIBRARY
58 Pm DECLARE FUNCTION GetMsg LIBRARY
59 lK DECLARE FUNCTION OpenWindow LIBRARY
60 mS DECLARE FUNCTION Examine% LIBRARY
61 37 DECLARE FUNCTION ViewAddress LIBRARY
62 sc DECLARE FUNCTION ViewPortAddress LIBRARY
63 BK DECLARE FUNCTION AutoRequest LIBRARY
64 lK0 premenne:
65 Rb2 nacitanie%=1
66 ho VypisPamete=0&
67 qp i%=0
68 Z3 x%=200

```

```

69 X1 y%=100
70 Kc lo1%=3
71 pL DIM SHARED dir.Meno$(x%)
72 bz DIM SHARED dir.Type$(x%)
73 4g DIM SHARED info$(x%),Tlac$(x%)
74 4y DIM a$(y%)
75 95 DIM a&(y%)
76 Kf0 okno:
77 f22 MacheRand FeldRand,-1,-1,1,150,14
78 W0 MacheRand AusgRand,-1,-1,3,630,28
79 lG MacheRand AusgRand1,-1,-1,3,260,14
80 bg IntText EditTxt,1,1,31,2,"Zmenit meno",0&
81 wu IntText WeiterTxt,1,1,23,2,"dalsie Data",0&
82 BE IntText OkTxt,3,1,35,2,"Prebratie",0&
83 ML IntText GoonTxt,1,1,27,2,"dalsia DIR",0&
84 wu IntText df0Txt,1,1,59,2,"df0:",0&
85 33 IntText df1Txt,1,1,59,2,"df1:",0&
86 At IntText DiskTxt,1,1,15,2,"dalsia Disketa",0&
87 pF IntText DruckTxt,1,1,15,2,"TLACENIE ETIKETY"
,0&
88 LU IntText FrageTxt,1,2,10,8,"Chcete naozaj
prestat?",0&
89 Jx IntText JaTxt,1,2,6,3,"Ano !!",0&
90 Ks IntText NeinTxt,1,2,6,3,"Nie ",0&
91 JX IntText DruckFrageTxt,1,2,10,8,SPACE$(11)+
"Vytlac ?",0&
92 EK FeldDef GoonFeld,0&,12,70,148,12,0,1,1,
FeldRand,GoonTxt,0&,1
93 7d FeldDef EditFeld,GoonFeld,167,70,148,12,0,1,
1,FeldRand,EditTxt,0&,2
94 0o FeldDef WeiterFeld,EditFeld,322,70,148,12,0,
1,1,FeldRand,WeiterTxt,0&,3
95 z3 FeldDef OkFeld,WeiterFeld,477,70,148,12,0,1,
1,FeldRand,OkTxt,0&,4
96 Zs FeldDef df0Feld,OkFeld,12,85,148,12,0,1,1,
FeldRand,df0Txt,0&,5
97 e4 FeldDef df1Feld,df0Feld,167,85,148,12,0,1,1,
FeldRand,df1Txt,0&,6
98 bB FeldDef DiskFeld,df1Feld,322,85,148,12,0,1,1,
FeldRand,DiskTxt,0&,7
99 2n FeldDef DruckFeld,DiskFeld,477,85,148,12,0,1,
1,FeldRand,DruckTxt,0&,8
100 di tituls$="<- KONIEC"
101 Et tituls$=tituls$+SPACE$(8)+"DiskEti V1.2 "
102 sf tituls$=tituls$+" modifikovany v 1989 M.Turekom"
103 tw OknoDEF N Fenster,0,0,639,100,64+512&,
8&+4096&,DruckFeld,tituls$
104 sY WINDOW CLOSE 2
105 ea WINDOW 3,,(0,99)-(629,186),0
106 LP AdresaOkna=OpenWindow(N Fenster)
107 lO IF AdresaOkna=0 THEN ERROR 7
108 P7 OknoRP=PEEK(L AdresaOkna+50)
109 Cc OknoUP=PEEK(L AdresaOkna+86)
110 so DrawBorder OknoRP,AusgRand,5,11
111 zQ DrawBorder OknoRP,AusgRand1,375,40
112 T0 DrawBorder OknoRP,AusgRand1,375,55
113 nD CALL SetAPen(OknoRP,1)
114 3M tlac1 "Data pripravene na spracovanie :",
9,41
115 Nd tlac1 "Prosim editujte aktualne datove
meno :",9,56
116 8v GOSUB UkazEtiketv
117 7e0 HlavnyProgram:
118 g02 GOSUB VstupPola
119 Y5 Vpln 6,11,631,36
120 SC tlac1 "Prosim vyberte si diskovu
jednotku...",9,19
121 Qu GOSUB Oznamenie
122 b8 Vpln 6,11,631,36
123 YQ0 CitajDir:
124 CD2 FOR slucka%=teraz% TO nacitanie%-1
125 Xf4 IF a&(slucka%)=stupen% THEN
126 DI6 hladaaj$a$(slucka%)
127 ek z%=x%
128 lG zeige%=0
129 iF Vpln 6,11,631,36

```

```

130 Se      tlc1 "Nacitavam :",9,11:tlc1 a$(slucka%)
,9,19
131 oc      tlc1 "Prosim trochu strpenia...",9,27
132 cB      CitajDisk hladaj$,x%
133 if      IF chyba%=500 THEN
134 TF8      novy%=1
135 Ta      GOTO esteraz
136 816     END IF
137 oE      max%=x%
138 v1      x%=z%
139 mY      Sort
140 Dd      IF slucka%=0 THEN
141 Ma8      a$(0)=dir.Meno$(0)+":"
142 OQ      Tlac$(0)=LEFT$(a$(0),LEN(a$(0))-1)
143 wT      Vypln 6,11,631,36
144 r3      tlc1 "Meno Diskety :",9,19
145 Yu      IF LEN(Tlac$(0))>15 THEN
146 4oA      Tlac$(0)=LEFT$(Tlac$(0),15)
147 JC8     END IF
148 Nr      tlc1 Tlac$(0),388,41
149 WD      ed$=Tlac$(0)
150 tN      GOSUB Oznamenie
151 r3      IF novy%=1 THEN GOTO esteraz
152 OH6     END IF
153 CC      z$=a$(slucka%)
154 i1      dir$=LEFT$(z$,LEN(z$)-1)
155 8f      Vypln 6,11,631,36
156 OL      tlc1 "Direktory :",9,11:tlc1 dir$,9,19
157 Pj      FOR anz%=1 TO max%
158 pm8      IF dir.Type$(anz%)<>"DIR" THEN
159 oYA      IF RIGHT$(dir.Meno$(anz%),5)<>
".info" THEN
160 uDC      IF RIGHT$(dir.Meno$(anz%),8)<>
".fastdir" THEN
161 mXE      info$(zeige%)=dir.Meno$(anz%)
162 Si      zeige%=zeige%+1
163 ZSC      END IF
164 aTA      END IF
165 Gz8      ELSE
166 rxA      fl=1
167 g3      a$(nacistanie%)=a$(slucka%)+dir.
Meno$(anz%)+"/"
168 q9      a$(nacistanie%)=stupen%+1
169 yS      nacistanie%=nacistanie%+1
170 g28      END IF
171 vq6      NEXT anz%
172 ib4      END IF
173 Oe      uloha%=0
174 jP      Spytanie:
175 hy6      IF uloha%<zeige% THEN
176 Wk8      info$(uloha%)=LEFT$(info$(uloha%),15)
177 FB      Vypln 380,40,631,50
178 aH      tlc1 info$(uloha%),388,41
179 Ng      ed$=info$(uloha%)
180 Nr      GOSUB Oznamenie
181 OW      IF novy%=1 OR tlacenie%=1 THEN GOTO
esteraz
182 hx      uloha%=uloha%+1
183 Sh      GOTO Spytanie
184 un6      END IF
185 u14      NEXT slucka%
186 x22      teraz%=slucka%
187 OG      IF fl=1 THEN
188 8D4      fl=0
189 sf      stupen%=stupen%+1
190 De      GOTO CitajDir
191 lu2      END IF
192 Ow0      esteraz:
193 2v2      IF novy%=0 THEN
194 bC4      Etikett
195 5y2      END IF
196 YU      Vypln 380,40,631,50
197 Gt      Vypln 9,84,317,97
198 21      GOSUB NovyInit
199 17      GOTO HlavnyProgram
200 Su0      Koniec:

```

```

201 Z62     CALL CloseWindow(AdresaOkna)
202 mU      CALL VolnaPamet
203 MX0      SCREEN CLOSE 2
204 ZG2      WINDOW CLOSE 3
205 x8      LIBRARY CLOSE
206 4z      END
207 jG0      Oznamenie:
208 KW2      lauf%=1
209 cE      Hlavnaslucka:
210 WJ4      IF lauf%=1 THEN
211 fK6      IntuiSprava=GetMsg(OknoUP)
212 kh      IF IntuiSprava>0 THEN GOSUB
IntuitionSprava
213 gr      GOTO Hlavnaslucka
214 OH4      END IF
215 pR2      RETURN
216 IH0      IntuitionSprava:
217 1s2      SpravaTyp=PEEK(IntuiSprava+20)
218 dw      Miesto=PEEK(IntuiSprava+28)
219 88      FeldNr%=PEEK(Miesto+39)
220 AE      CALL ReplyMsg$(IntuiSprava)
221 UG      IF (SpravaTyp=64) THEN
222 XN4      IF FeldNr%=1 THEN
223 Wh6      lauf%=0
224 Bl      uloha%=zeige%
225 wx4      ELSEIF FeldNr%=2 THEN
226 UK6      StrInfo EString,ed$
227 YD      FeldDef StrGad,0&,389,57,150,12,0,1,
4,0&,0&,EString,9
228 Rs      CALL AddGadget(AdresaOkna,StrGad,-1)
229 qa      CALL RefreshGadgets(StrGad,AdresaOkna,0&)
230 H2      CALL ActivateGadget(StrGad,AdresaOkna,0&)
231 464      ELSEIF FeldNr%=3 THEN
232 fq6      lauf%=0
233 8B4      ELSEIF FeldNr%=4 THEN
234 ed6      IF Tlac%>0 THEN
235 av8      Tlac$(Tlac%)=info$(uloha%)
236 kd6      END IF
237 VS      GOSUB PrintOut
238 9o      Tlac%=Tlac%+1
239 mx      lauf%=0
240 HT      IF Tlac%=33 THEN tlacenie%=1
241 8x4      ELSEIF FeldNr%=5 THEN
242 pO6      lauf%=0
243 RV      a$(0)="df0:"
244 vu      GOSUB PolePrec
245 OT4      ELSEIF FeldNr%=6 THEN
246 t46      lauf%=0
247 bb      a$(0)="df1:"
248 zy      GOSUB PolePrec
249 Ua4      ELSEIF FeldNr%=7 THEN
250 L76      novy%=1
251 y9      lauf%=0
252 Zg4      ELSEIF FeldNr%=8 THEN
253 OB6      lauf%=0
254 1J      tlacenie%=1
255 em4      ELSEIF FeldNr%=9 THEN
256 4i6      CitajEdit EString,r$
257 xq      IF Tlac%<1 THEN
258 WE8      Tlac$(0)=r$
259 ZV      Vypln 380,40,631,50
260 Bf      tlc1 Tlac$(0),388,41
261 N9      CALL RemoveGadget(AdresaOkna,StrGad)
262 H6      CALL RefreshGadgets(GoonFeld,
AdresaOkna,0&)
263 qZ6      ELSE
264 b38      info$(uloha%)=r$
265 fb      Vypln 380,40,631,50
266 Oh      tlc1 info$(uloha%),388,41
267 TF      CALL RemoveGadget(AdresaOkna,StrGad)
268 NC      CALL RefreshGadgets(GoonFeld,
AdresaOkna,0&)
269 HA6      END IF
270 zD      Vypln 376,55,631,66

```

Dokončenie listingu programu nájdete v budúcom čísle.

INZERCIA

■ Firma GraTex predá za zostatkovú cenu polročný počítač AT386SX, 16MHz, mat. koprocessor 80387SX, 8Mb RAM, 96Mb HD (25ms), 1.2" + 1.44" FD, 2 ser./2 par., VGA 1024x768, 512Kb, 16 bit, komplet s VGA monochrom. monitorom za 105.000,- Kčs, resp. s profesionálnym NEC VGA Multisync 3D 1024x768 za 125.000,-. V cenách je započítaná aj daň z obratu. GraTex, Zaduňajská 7, 851 01 Bratislava

■ Pro Commodore C64/128 prodám Turbo Cartridge "TIP-TOP" s reset tlačítkem pro uživatele disku. Mimo jiné obsahuje 6 x rychlejší load a komfortní DOS. 100 % stavi vzhled. Záruka. Zašlu i na dobírku. Jiří Krivický, Přímětická 1202, 14000 Praha 4

■ Majiteľom tlačiarne SP 180VC (pre C-64, C-128) ponúkam upravenú EPROM so všetkými slovenskými a českými písmenami, veľké i malé, niektoré symboly gréckej abecedy, popis tabuľky znakov. Znaková sada Commodore je zachovaná, upravená len znaková sada DIN. Stačí zaslať pôvodnú EPROM a obratom získate úpravu (á 250 Kčs). Záruka, spoľahlivosť. Ing. Tokár Miloslav, Sov. armády 55/3, 92701 Šaľa

■ Predám IO 74LS123 (22), 74LS157 (20), 74LS173 (24), 74LS244 (26), 74LS273 (25), 74LS373 (27), 74LS374 (29), EPROM 27C128 (190), Z80APIO (80), ACTC (80), ASIO-O (160), ADART (220), plast. stab. 78M08 0,5 A (10), 7808 1,5A (15), Tranzistory BFT66 (110), BFQ69 (120), osciloskop do 10 Mhz C1-94 nový (2400), literatúru pre Commodore 64, Ing. E. Dzurányi, Jégého 7, 82108 Bratislava, Tel. 07/65469

■ Hľadám majiteľov programov kompatibilných s Geosom, programy na výmenu, predám tlačiareň Gamacentrum 01 (1600 Kčs), nová nepoužitá, vhodná pro PMD, ATARI, COMMO-

Máte strach z počítačových vírusov?

ANTIVÍRUSOVÝ PROGRAM



NOD

zachytáva vyše 200 variánt vírusov
a ponúka komplexnú ochranu
systémových oblastí PC
v integrovanom prostredí

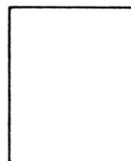
A.T.A. konzorcium

Tel: 0805/201 98, Fax: 0805/205 17
Jerichov 14, 917 01 TRNAVA

Cena: 2 600 Kčs

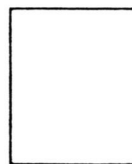
heslo:

"NOD"



*redakcia časopisu
POČÍTAČ aktívne
P. O. Box 23
835 32 Bratislava*

ERKA
press



*ERKA press
P. O. Box 23
835 32 Bratislava*

Tipy a triky pre Commodore 128 (D)

Tipy a Triky pre počítač C-128 (D). Prinášame niekoľko POKE-ov a PEEK-ov a niekoľko zaujímavostí pre diskovú jednotku 1571. Takže, nech sa páči.

Použili ste omylom príkaz NEW, alebo ste museli vykonať RESET lebo počítač "zamrzol" a váš program v jazyku BASIC nemožno opäť spustiť. Tu je riešenie:

```
POKE 1+256*PEEK(46),1:RENUMBER
a program je opäť na svete.
```

Príkazom SYS 65378 zavediete späť originálnu znakovú sadu procesora VDC.

Otázka na práve prevádzaný riadok jazyka BASIC znie:

```
PRINT PEEK(59)+256*PEEK(60)
```

Otázka na aktuálny DATA riadok znie:

```
PRINT PEEK(65)+256*PEEK(66)
```

Počet znakov v zásobníku klávesnice 128-ky zistíme príkazom:

```
PRINT PEEK(208)
```

Príkaz

```
POKE 774,38:POKE775,160
```

spôsobí, že po zadaní príkazu LIST sa budú na obrazovke ukazovať len čísla jednotlivých riadkov programu. Toto

môže slúžiť ako jednoduchá ochrana pred nevítanými hosťami. Profíkov však nezastaví.

Príkaz

```
POKE 774,61:POKE 775,255
```

prevedie pri pokuse o LIST - PROGRAM RESET počítača.

Príkaz

```
POKE 808, PEEK(808)-3
```

spôsobí odpojenie klávesy RUN/STOP a kombinácie kláves RUN/STOP a RESTORE.

```
SYS 49194
```

prepína 40 a 80 znakový modus výstupu pre monitor.

```
SYS 65357
```

alebo

```
SYS 57931
```

spôsobí priamy skok do C64 modu.

Jednostranne naformátované diskety z disketovej jednotky 1541 sa na jednotke 1571 dajú čítať po prevedení príkazov:

```
OPEN 15,8,15,"u0>m0"
PRINT#15,"m-w",chr$(172),
chr$(2),chr$(1),chr$(41)
CLOSE 15
```

Doteraz sa vždy vyskytovali problémy, ak program používal sprity (sprite - čítaj "sprajt") zároveň pracoval s disketovou jednotkou. Problém bol v tom, že pri čítaní z diskety začali vždy sprity blikať. Tomuto sa dá predísť nasledujúcimi príkazmi jazyka BASIC:

```
SPRITE 1,1,2
MOVSPR 1,100,100
DIRECTORY
```

Veľmi nepríjemné je toto blikanie hlavne v tých BASIC-ových programoch, ktorých sprity pracujú so 40-znakovou obrazovkou, kde sa blikanie prejaví v prípade zásahu na disketu. Riešenie problému je nasledovné:

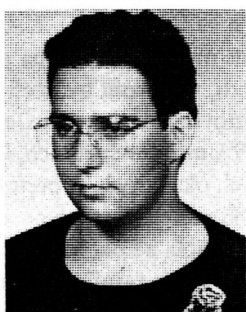
Pamäťová bunka s názvom "holdof" (\$0A3A) v systémovej pamäti počítača má bitom 7 určené, či je neustále zapínanie a vypínanie sprítov povolené (0), alebo povolené nie je (1). Po zadaní

```
POKE 2618,128
```

sa už toto nepríjemné blikanie sprítov pri zásahu na disketu neobjaví. Pôvodný stav získame príkazom:

```
POKE 2618,0
```

AMIGA



Vážení čitatelia !

Dovoľte mi, aby som Vás podrobnejšie informoval o vývoji rubriky Commodore AMIGA v našom časopise. Túto rubriku už začali ovplyvňovať prvé telefonáty a listy, takže by ste si aj Vy mali nájsť odpoveď na Vaše otázky v našich článkoch. Nemusíte sa teda báť, že sa staneme plnoautomatizovaným prekladacím zariadením časopisu Amiga od firmy Markt & Technik, z ktorého máme povolenie preberať informácie. Chceme však čerpať hlavne programy a popisy k nim a venovať sa problémom aktuálnych pre našich užívateľov.

V ďalších číslach sa budeme zameriavať predovšetkým na kurzy pre začiatočníkov, ktorí si už v nasledujúcom čísle nájdú svoj kurz práce s CLI. No nezabúdame ani na pokročilých a profesionálov, pre ktorých chystáme kurz jazyka C a assembleru. Z priestorových dôvodov budú vynechané dlhé listingy programov a budú uverejňované len tie kratšie. No nemusíte sa báť, že uverejnený popis k programu bude pre Vás zbytočný, lebo listing, alebo už disketu s programom, si môžete za symbolickú cenu objednať na našej adrese. Predpokladáme totiž, že mnoho programov, ktorých popis uverejníme, máte doma, ale bez popisu. Takže, kto má program, nájde k nemu u nás popis a kto nemá program, tak si ho môže objednať. Budeme distribuovať len tie programy, u ktorých to bude výslovne uvedené. Viac sa budeme orientovať na popisy základných užívateľských, grafických a hudobných programov.

V dnešnom čísle nájdete druhú časť článku Dejiny jedného úspechu. V tipoch a trikoch sa dočítate o tom ako môžete prepojiť váš počítač AMIGA s televízorom alebo videom bez toho, aby ste potrebovali TV modulátor. Tento problém operatívne riešime na základe prvých telefonátov. Ďalší problém, ktorý nám nastolili čitatelia, je problematika pripojenia tlačiarň, o ktorej sa zmienime v ďalšom čísle.

Pre tých čo nedajú dopustiť na počítačové hry je pripravený popis známej hry Prepadisko. Listing si môžete objednať podľa podmienok, ktoré sú uvedené na inom mieste. Uverejňovať sa budú i rôzne popisy a návody k hrám, tipy a triky, ako napr.: nesmrteľnosť, kódy a plány miestností, atď.

Naša redakcia vypisuje súťaž o najlepši príspevok rubriky AMIGA. Vaše príspevky budú postupne uverejňované a autor toho najlepšieho bude odmenený hodnotnou cenou ako napr.: knihou, alebo softwarom od firmy Markt & Technik. Príspevky môžu byť z najrozličnejších oblastí hardwaru a softwaru, ale i rôzne návody a tipy k hrám.

Chcel by som Vás požiadať, aby ste nám svoje pripomienky a dotazy posielali na adresu redakcie časopisu Počítač aktívne. Pomôžete nám tak v ďalšej orientácii časti venovanej Amige.

Preto neváhajte a píšite.

VLADIMÍR ORLÍK

Vladimír

Orlík

Dejiny jedného úspechu II.

Amiga oslavuje svoje piate narodeniny. Časopis AMIGA informuje o svojich začiatkoch až po najnovšie modely.

V júli 1985 bola Amiga 1000 oficiálne predstavená v newyorskej opere. Odvtedy už víťazný postup Amigy nemožno zastaviť. V tomto článku sa dozvieme, ako sa z Amigy 1000 stala Amiga 2000 (model-A), ako vznikol Sidecar a prvá karta (A 2088) a aký mala účinok na vývoj Amigy.

Otočme koleso dejín Amigy naspäť do roku 1985. So zavedením Amigy 1000 na trh sa začali vo vývojovom oddelení firmy Commodore v Braunschweigu zaberať myšlienkou, ako by tomuto počítaču dali kompatibilitu so systémom MS-DOS. Od tohto konceptu si sľubovali úplnú schopnosť multitaskingu zapojením fantastických možností zvuku a grafiky Amigy. Prvé koncepty poskytli síce prijateľné riešenia, boli však už vopred odsúdené na zánik, na základe príliš vysokých vývojových nákladov pri vtedajších trhových cenách. Irving Gould, prezident a väčšinový akcionár firmy Commodore, poveril 10. marca 1986 dr. Henryho Rubina ako osobného poradcu, aby vyriešil finančné a technické problémy. Dr. Henry Rubin bol dovtedy šéfom veľkej vlastnej obchodnej spoločnosti v Južnej Afrike. Henry Rubin prices-toval do Braunschweigu s jasným cieľom nájsť cestu, ako cenovo výhodne vyrobiť Amigu 1000, aby bola kompatibilná s MS-DOS. Pre vtedajšiu skupinu vývojových pracovníkov v Braunschweigu to bol tvrdý oriešok. Henry Rubin však vyhlásil, že do šiestich týždňov musí byť vyvinutý použiteľný výrobok, aby mohol urobiť Amigu 1000 kompatibilnou s MS-DOS.

Pracovníci vývoja sa pustili do boja. Pripomeňme si: všetko, čo sa týkalo Amigy, robili vlastní vývojoví pracovníci, ako Jay Miner, ktorý je uznávaný ako

praotec Amigy. Frank Ullmann (Commodore Braunschweig) naozaj vyvinul za šesť týždňov počítač kompatibilný s PC, ktorý sa dal napojiť na Amigu 1000. Pozostával z troch plošných spojov a bol zabudovaný s materskou doskou Amigy 1000 do obrovskej PC-skrinky. Potrebný Software-Interface skonštruoval Dieter Preiss (dnes technický riaditeľ firmy Commodore, Nemecko). V pôvodnej verzii disponoval len jedným, presne umiestneným oknom na počítači Amiga, ktoré mohlo zobrazovať normálny textový display PC-čka. Túto konštrukciu - za dodržania stanovenej lehoty - predstavili o 6 týždňov zodpovednému grémiu firmy Commodore v New Yorku. Všetci spolupodpovední boli nadšení.

Zdalo sa, že vytúžený cieľ - prísť na trh s počítačom, ktorý by popri vynikajúcich grafických a zvukovotechnických vlastnostiach umožňoval naozajstnú multitaskingu a multiprocessorovú prevádzku, je dosiahnutý.

Sidecar - prvý PC pre počítač Amiga

Trpká príchuť však zostala. S predloženým konceptom by výrobné náklady enormne stúpili. Takže najskôr bolo treba hľadať riešenie, ktoré by podporilo koncept pripojiteľného hardwaru s integrovaným softwarom, aby Amiga 1000 nebola kompatibilná len s MS-DOS, ale aj konkurencie schopná. Preto vyvinuli z existujúceho prototypu prístroj, ktorý sa mal zapísať do dejín ako "Sidecar". Dal sa napojiť na každý počítač typu Amiga 1000 a tvoril prepojenie (interface) počítačov Amiga a sveta MS-DOS a naopak. Názov Sidecar bol najprv inter-

ným označením (Codename). Aby sa projekt uskutočnil, príleteli do Nemecka ľudia, ktorí už od začiatku spolupracovali na vývoji počítača Amiga. Treba spomenúť najmä Boba "Kodiaka" Burnsa a =RJ= Mical (obe "sú jeho poznávacie znamienka, ktoré sú fanúšikom počítača Amiga známe ako "Gurus").

Aby ponúkli vývojovým pracovníkom počítača Amiga v Nemecku adekvátne pracovisko, musela si vtedajšia skupina šéfov z Braunschweigu poriadne lámať hlavu (a kadečo zničiť). Kalifornčania, zvyknutí na slnko, trvali na pracovnom prostredí, v ktorom by nechýbalo nič z ich domoviny, čo znamenalo, že chodiť do služby mohli v šortkách, teniskách a slnečných okuliarech. A pretože obidvaja vývojoví pracovníci trpeli posunom času, bolo to zväčša až okolo obeda. Do pracovne im zaobstarali tropické rastliny, ústredné kúrenie pustili naplno (marec 1986 nebol najteplejší) a zrušili všetky osobné smernice, ktoré vtedy platili pre nemeckých zamestnancov. A tak Bob Kodiak Burns a =RJ= Mical spolu s nemeckými vývojovými pracovníkmi Torstenom Burgdorffom a Dieterom Preissom vybojovali "bitky o software", z ktorých dodnes ťaží každý majiteľ a užívateľ počítača Amiga 1000 so Sidecarom alebo počítača Amiga 2000 s kartou, aj keď bol software dodnes viackrát prepracovaný. Bob napísal prvú verziu "Janus-Library", "DJ Mount" a prerušovač (LPT1), =RJ= Mical vyvinul "PC Windows". Od Dietera Preissa pochádzal JDISK.SYS a JLINK.COM a Torsten Burgdorff vyvinul Janus Software-Interface medzi počítačom Amiga a PC. Keď Sidecar splnil všetky očakávania horeuvedeného managementu, nedal si prezident Irving Gould ujsť návštevu tohto sympatického veľkomesta Henricha Leva (Braunschweig) a jeho vývojového tímu.

Návšteva vyvrcholila pohostením so zúčastnenými vývojovými pracovníkmi v jednom z najlepších zariadení (Haus zur Hanse), ktoré mesto mohlo vtedy

ponúknuť. Po jedle si dal prezident pri-niesť nápojový lístok. Na lístku objavil koktail, ktorý mal meno "Sidecar". Prekva-pený touto zhodou, rozkázal pán Gould jedno kolo Sidecaru pre všetkých.

Koncom apríla 1986 Sidecar prvý raz verejne predstavil na Comdaxe. V rov-nakom čase vyvinuli pracovníci okolo Joyho Mineru nový počítač Amiga, ktorý

nazvali Amiga 2000. Tento počítač mal naozaj objemnú skrinku a disponoval obmedzeným bus-konceptom. Vtedajšie-mu vedeniu firmy Commodore to však nestačilo. A tak dostali vývojoví pracov-níci v Braunschweigu za úlohu vyvinúť počítač Amiga, ktorý by disponoval takým slotkonceptom, aby umožnil umiestniť v prístroji aj PC-karty.

Amiga 2000 (model A) je na svete

Paralelne k tomuto počítaču sa mala vy-viniť zasúvací karta, kompatibilná s XT. Podklad pre nový počítač Amiga predstavovala Amiga 1000. Bol opäť prepracovaný a doplnený novým slotkon-ceptom.

Nový počítač nazvali v Commodore "Amiga 2500". Jeho zvláštnosti nájdete po prvýkrát popísané v tomto príspevku. Pretože počítač Amiga 2000, vyvinutý Joyom Minerom, napokon predsa len nezostrojili, dostal toto pomenovanie počítač, vyvinutý v Braunschweigu. Aby sa mohla urobiť zo Sidecaru zasúvací karta pre nový počítač Amiga, bolo treba jeho veľkosť zredukovať. Pre tento účel vyvinul Bernd Asmann (vývojový pra-covník Commodore v Braunschweigu) ABT (Adress Bus Translator) a DBT (Data Bus Translator). S novovyvinutým čipom sa ušetrilo oproti starému Side-caru cca 40 čipov. Z prepracovaného počítača Amiga 1000 vznikla prvá Amiga 2000 (model A) a zo Sidecaru, takisto vyvinutého v Braunschweigu vznikla, T-karta. Ďalšiu generáciu potom vytvorili vo West Chesteri (USA). Z nej vzišiel počítač Amiga 500 a nová verzia počítača Amiga 2000 (model B).

V niektorom z ďalších vydaní čísel časopisu budeme informovať o vývojo-vých pracovníkoch z West Chesteru, ktorí postavili na nohy počítač Amiga 3000. Okrem toho nazrieme aj do bu-dúcnosti, čo nového sa plánuje pod znač-kou Amiga.

(AMIGA, 8/90, spracované)

Nová verzia
AMIGA 2500/30
od firmy Commodore.



Príchut' Bratislavy

Pôvodne sme chceli na tomto mieste priniesť informáciu z najväčšieho podujatia v oblasti počítačov typu Amiga na americkom kontinente - veľtrhu v Chicagu v lete minulého roku. Potom sme si povedali, že do Bratislavy je predsa len trochu bližšie ako na sever USA a hlavne, že aj Bratislava je miestom zaujímavej aktivity okolo počítačov Amiga. Navštívili sme totiž Amiga - klub Slovakia a urobili rozhovor s jeho predsedom Zoranom Nikoličom.

Redakcia: Pravdepodobne väčšina čitateľov držiacich v tomto momente druhé vydanie časopisu "Počítač aktívne" určite počula aj o Vašej činnosti, takže poďme hneď k veci. Čo je Amiga Club Slovakia?

Z.N.: Amiga Club Slovakia alebo ACS je voľné združenie priateľov, užívateľov a iných nadšencov pre počítače Amiga s celoslovenskou pôsobnosťou.

Redakcia: Po tejto priveľmi stručnej odpovedi by sa čitatelia určite radi dozvedeli trochu viac o vašej náplni, vašom celkovom zameraní, cieľoch činnosti apod.

Z.N.: Členom klubu samozrejme poskytujeme trochu viac. Je to v prvom rade voľný prístup na prednášky a semináre zo širokého výberu tematických oblastí a poradenská služba na hardware aj software. Ďalej ponúkame servis počítačov a prídavných zariadení, prístup k odbornej literatúre a k informáciám o aktuálnych programoch. Sprostredkujeme lacný nákup výpočtovej techniky a doplnkov pre počítače Amiga. A v neposlednom rade sa naša aktivita koncentruje na organizovanie počítačových výstav a stretnutí pre širokú verejnosť. Myslíme aj na tých najmenších, pre ktorých pripravujeme zábavné posedenia, plné počítačových hier.

Redakcia: Priznám sa, že si ma trochu zaskočil. Vaša činnosť je naozaj vskutku bohatá. Ale predstavme si, že som čerstvým majiteľom Amigy 500, nikdy som sa predtým nedostal do kontaktu s výpočtovou technikou a dozviem sa o Amiga klube Slovakia. Prídem ku vám a čo ďalej?

Z.N.: Postup je jednoduchý. Najprv ti doporučím, aby si sa stal členom ACS - kvôli už spomínaným akciám a ďalším výhodám, o ktorých ešte len bude reč. Iste budeš mať záujem o pravidelný kurz pre začiatočníkov. Neskôr sa môžeš zdokonaľovať v ľubovoľných odboroch podľa vlastných schopností.

Redakcia: Ďakujem za ponuku, stojím za úváženie, ale zrejme všetko niečo stojí, takže čo za to?

Z.N.: Samozrejme, že všetko niečo stojí a naša snaha vedie k tomu, aby výdavky pre člena boli čo najmenšie. Našli sme rôzne cesty financovania našich podujatí od zaujímavých a ochotných sponzorov. Radi by sme ale našli možnosti na financovanie celej našej činnosti, čo v preklade znamená minimálnu záťaž na peňaženku člena pri náraste profesionálnej orientácie ACS.

Redakcia: Amiga a s ňou kúzelný svet hier, farieb a hudby k nám prišla od západných susedov. Plánujete vo svojej činnosti nadviazať užší kontakt s podobne zameranými organizáciami na západe?

Z.N.: Som rád, že si sa dotkol tejto témy. Kontakty sú veľmi dôležitá vec. Podarilo sa nám nadviazať úzke spojenie s Amiga klubom Österreich. Na júl plánujeme medzinárodné stretnutie amigistov, kde predpokladáme, že sa nám podarí zabezpečiť nadviazanie spolupráce s ďalšími svetovými klubmi podobného zamerania.

Redakcia: Plány sú jedna vec a realizácia zámerov iná. Čo napríklad môžeš ponúknuť zahraničným partnerským klubom, aby si prilákal ich reprezentantov na našu pôdu?

Z.N.: Pokladáme to za najvýznamnejšiu tohtoročnú akciu, preto jej prikladáme aj náležitú pozornosť. Všetky kroky sta-

roslivo pripravujeme vo výbere ACS a zapojíme do spolupráce aj organizácie, s ktorými sme v úzkom kontakte. Radi prijmem aj ďalšie podnetné návrhy a pomoc.

Redakcia: Ako Amiga-nadšenec sa už vopred teším na toto podujatie. Mohol by si konkretizovať ďalšie podmienky, to znamená s akou návštevnosťou počítate, kto sa môže tohto podujatia zúčastniť a kde sa bude celé podujatie konať?

Z.N.: Dovolil by som si začať od konca. Podujatie sa bude konať v priestoroch Slovenského domu detí a mládeže v Bratislave. Zúčastniť sa ho môže každý člen ACS, pre nečlenov bude účasť s príplatkom a ubytovanie budeme zabezpečovať len pre zahraničných hostí a mimobratislavských členov ACS. Návštevnosť si zatiaľ netrúfam odhadnúť, ale budeme klásť dôraz predovšetkým na kvalitu podujatia.

Redakcia: Doposiaľ sme hovorili o plánoch ACS. Môžeš sa pochváliť aj realizovanými podujatiami?

Z.N.: Určite. Okrem nami zorganizovaných výstav so zameraním na výpočtovú techniku sme sa spoločne zúčastnili na výstave "AMIGA WORLD" vo Viedni. Vzhľadom na veľký záujem máme zorganizovaný spoločný zájazd na najväčšiu medzinárodnú výstavu počítačov Amiga v Berlíne koncom apríla. V spolupráci so spoločnosťou AMIUM prebehla pre členov ACS súťaž o najkrajší grafický námet s hodnotnými cenami a pripravujeme veľkú súťaž pre záujemcov z oblasti počítačovej animácie a hudby.

Redakcia: Naš redakčný kolektív samozrejme teší aktivita tohto druhu a dúfame, že v blízkej budúcnosti nadviažeme užšiu spoluprácu. Ďakujem ti za rozhovor a prajem ti splnenie všetkých plánov.

Z.N.: Ja Vám tiež ďakujem. Pre záujemcov ešte uvediem to podstatné:

Naša adresa:

Amiga Club Slovakia, Mehringova 4, (Mynarovičova 4), 851 04 Bratislava alebo osobne v priestoroch ACS (každý deň 17.00 až 21.00 h): Slovenský dom detí a mládeže Trieda Laca Novomeského 64, 842 58 Bratislava.

Prepadlisko

Prepadlisko je hra, využívajúca doslova dvojité podlahu.

Ak ju otvoríte, pripravíte pre svojho protivníka pád do prepadliska.

Hracie pole pozostáva z 36 políček. Každé políčko môže mať dve vrstvy. Dôležité je vedieť, že biela leží nad oranžovou, čo vedie k mnohým úskaliam. Pri pohľade na hracie pole sú vľavo umiestnené posuvacie kolíky pre biele podložie a v hornej časti pre oranžové. Premiestňovaním kolíkov pre príslušnú farbu podložia zakrývame čierne diery prepadliska alebo otvárame pascu pre súpera. Ak hracia guľička leží na poli, ktoré sa zdalo byť celkom bezpečné, tak posunutím kolíka môže dôjsť k ohrozeniu súpera. Figeľ je v tom, že vzhľadom na toto usporiadanie nie je jasné, či pod bielym je ešte oranžové podlažie alebo vysunutím bieleho kolíka spadne guľička priamo do prepadliska. Jedno je isté: Ak stojíte na oranžovom poli, vysunutím kolíka spadnete do pasce. Avšak hra nie je stratená, pretože každý hráč má k dispozícii minimálne 5 guľičiek, takže pri prvom páde nie ste nevyhnutne stratený.

Na začiatku sa program pýta na počet a meno hráčov a potom v striedavom poradí si jednotliví hráči rozmiestňujú svoje guľičky na hracom poli. Ak sú všetky guľičky rozmiestnené, začína prvý hráč s prvým pokusom o zhodenie protihráča do prepadliska. K tomu slúži taktika ťahania a zasúvania kolíkov pre biele a oranžové podložie. Akonáhle sa podarí odkryť pod súperom pascu, ten sa do nej nenávratne prepadne. Program volá hráčov striedavo v poradí a nie je možné ťahať dvakrát za sebou. Ak už hráč nemá hraciu guľičku v poli, prehráva. Hra je ukončená, akonáhle zostal posledný hráč v poli. Ak končia v pasci guľičky súperov v jednom kole, hra zostáva nerozhodná.

Po ukončení hry sa program spýta, či chcete hrať ešte raz. Ak nie, treba stlačiť <ESC> a nasledovne <RETURN>. Po spustení Basicu naštartujte program obvyklým príkazom OPEN a RUN.

Z dôvodu maximalizácie informácií na minime priestoru neuverejňujeme listing tejto hry vnútri čísla. Všetci tí, ktorí si chcú hru zahrať si však jej listing môžu objednať na adrese redakcie a natukáť do počítača pomocou programu Checksummer, ktorý bol uverejnený v marcovom čísle. Čitateľom s nedostatkom času pošleme program priamo na diskete. (Bližšie pozri podmienky pre distribúciu programov na strane 48)

Elvíra

SUPER HRA POSLEDNÝCH MESIACOV

Na trh hier sa pred niekoľkými mesiacmi dostala jedna z najlepších hier posledného obdobia - ELVIRA.

Niekoľko slov na úvod: Nosí vlasy vyčesané do hora ako žena z domácnosti v 50. rokoch. No napriek tomu je zmiešaninou sexbomby Dolly Parton a nepríjemnej vampirelly Femme - americká hororová postava Elvíra. Popularita z filmu a televízie jej napomáha pri vstupe do sveta počítačov. Hra nosí jej meno s podnázvom "Mistress of the Dark" - Vládkyňa temnôt. Je zmesou dobrodružnej a akčnej hry. Hráč zažije dobrodružstvo aj bez toho, aby musel zadávať text - stačí iba vybrať si z menu. Elvíra zdedila po svojom zosnulom strýkovi zámok, v ktorom už dávno býva lady Emelda, umelkyňa v čiernej mágii, ktorá sa snaží prevziať vládu nad svetom. Je iba jediná možnosť ako jej úmysel prekaziť: Treba použiť protikúzlo, ktoré je skryté v rakve. Tá je zamknutá šiestimi zámkami. No Elvíra sa zdá byť

príveľmi mierna, alebo nie až tak múdra, aby mohla túto ťažkú úlohu splniť sama. Preto jej musíte pomôcť. Musíte zabiť démonov, ktorí obývajú zámok, aby ste získali potrebné kľúče. Samozrejme, nie je také jednoduché preklenúť uzavretý kruh, ktorý je zložený z vlka, vampíra, kostlivca a ľudožrúta. Nesmie sa zabudnúť ani na tlstú kuchárku so sekáčom. Tí všetci vás chcú rozseknúť, sťat' vám hlavu, vysať krv alebo ináč si na vás zgustnúť.

Tieto scény, ktoré su veľmi dobre graficky spracované, robia z hry pravý filmový horor, nevhodný pre deti. Hrad a okolie sú robené v 3D-grafike, v ktorej sa dá veľmi ľahko orientovať. Veľmi zaujímavá je scéna, keď sa mečom zabíjajú obyvatelia hradu. Viac sily a obratnosti pre tieto boje poskytujú čarovné nápoje, ktoré vám Elvíra priniesla do kuchyne. Ale samozrejme, najprv treba nájsť 47 správnych prísad. Je to obsiahla úloha pre pravých špecialistov na dobrodružné hry.

Praktické zapojenia

VČSFR je veľa užívateľov, ktorí nevlastnia monitor. Preto v tomto čísle uverejňujeme jednu z možností ako pripojiť Amigu k televízoru, bez toho aby bolo potrebné zakúpiť TV-modulátor, ktorý stojí okolo päťsto šilingov.

No má to jeden háčik. V televízore, ku ktorému chcete pripojiť vašu Amigu musí byť zabudovaný SCART-konektor. (Schéma zapojenia je na obr. č. 1)

Vysvetlivky:

Miesto označené na obrázku "1." nahradíte 470 ohmovým odporom a miesto označené "2." nahradíte 56 ohmovým odporom.

Zoznam súčiastok:

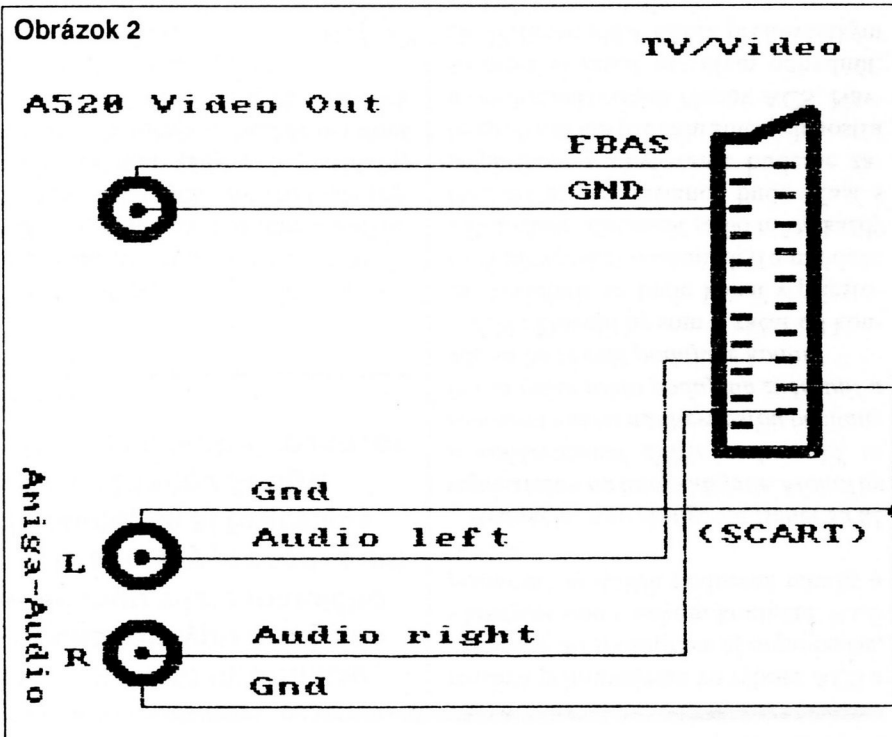
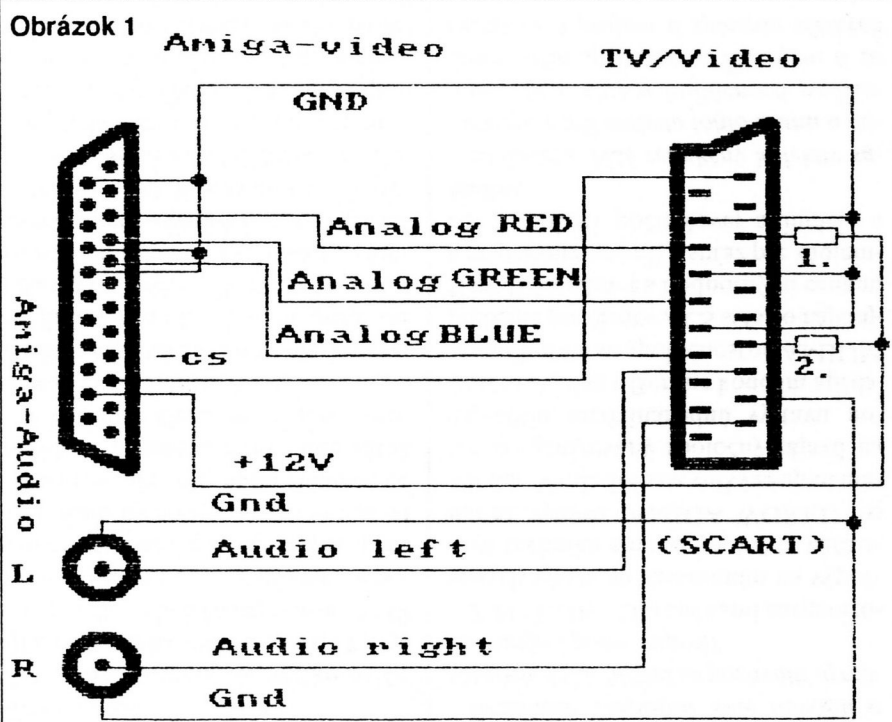
- 1x470 ohmový odpor
- 1x56 ohmový odpor
- 1x23-pólový konektor
- 1xSCART-konektor
- 2xCINCH-konektor

Niektoré televízory a videá aj napriek tomu, že majú SCART-konektor, nemôžu prijímať analógový RGB-signal.

Tieto zariadenia využívajú FBAS-signal. Z tohoto dôvodu vám ponúkame ďalšiu schému (obr. č. 2), pomocou ktorej bude prepojenie vášho počítača s TV, alebo videom úplnou hračkou.

Zoznam súčiastok:

- 1xTV-modulátor
- 1xSCART-konektor
- 3xCINCH-konektor



Čarovný svet fraktálov

Ten, kto sa trochu bližšie zaoberal experimentálnou matematikou, vie o čo ide - suchý svet čísiel, stvárnený do kúzelných útvarov, pripomínajúcich okrúhle postavičky akoby zaodeté do čipkovej výzdoby. Z iného pohľadu - fantastické grafické výstupy na obrazovke monitora.

Fraktálna grafika patrí do kategórie obrazovo pretavených matematických algoritmov pri zachovaní voľnosti tvaru a farebnej koncepcie. Tvary tohto typu počítačovej grafiky už dávnejšie zaujali umelcov, ktorí majú blízko ku klávesnici počítača a z jednoduchých symetrických útvarov (Kochove snehové vločky alebo počítačom generované stromy) sa stala nová kategória počítačového umenia, využívajúce náročné, počítačom realizované výpočty. Odvážnemu začiatočníkovi, ale aj náročnejšiemu užívateľovi poskytneme základy na experimentovanie v tejto oblasti. Budeme sa zaoberať pojmami ako 'fraktál', 'fraktálny objekt' a

poskytneme programy, ktoré čitateľovi umožnia vstup do čarovného sveta fraktálnej grafiky, realizovanej na počítačoch Amiga.

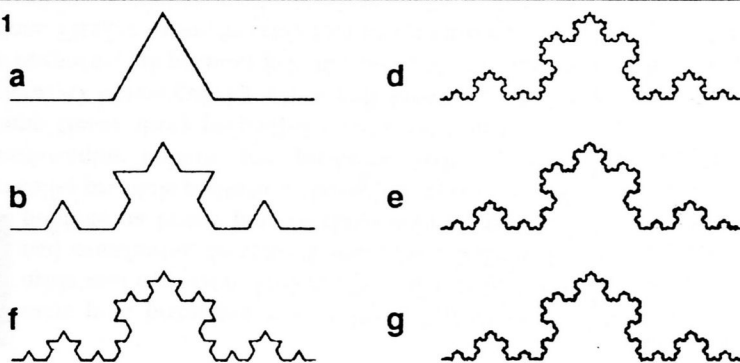
Celé sa to začalo v roku 1980 v Amerike, kde matematik Benoit B. Mandelbrot, zaoberajúci sa rôzne symetrickými geometrickými tvarmi, prišiel na množinu čísiel, odvtedy známu ako Mandelbrotova množina. Pri grafickom zobrazení parametrov Mandelbrotovej množiny čísiel vznikajú charakteristické obrazy, pripomínajúce figúrky akýchsi postavičiek alebo vzor dokonale prepracovanej čipky. Pri hľadaní pojmu sa udomácnil Mandelbrotov výraz 'fraktál' (anglicky - fractal) z latinského prídavného mena

'fractus' (zlomený), pretože vytvárame nepravidelné útvary. (Všeobecne sa ustálil pojem Mandelbrotova grafika, slangovo mandelbrotka - pozn. prekl.) Celý vývoj fraktálnej grafiky dostal tempo s vývojom zobrazovacích možností modernej výpočtovej techniky. Dnes už existuje celé odvetvie fraktálnej grafiky, idúce niekedy až na časové hranice možností výpočtovej techniky. Čomu vlastne vďaka toto odvetvie za taký živý záujem? Je to hlavne v obrovskej možnosti vytvárania nových, nesmierne zaujímavých tvarov. Hlavne však nový výpočet prináša vždy nový úkaz a pre tvorcu je nesmierne napínavé sledovať, čo mu počítač vykreslí pri danej definícii počiatočných podmienok. Fraktálna geometria môže popisovať také rozmanité javy, ako je rast kryštálov alebo vytváranie nových polymérov. Stala sa už celkom prirodzeným vyjadrovacím prostriedkom pre celý rad vedeckých oblastí.

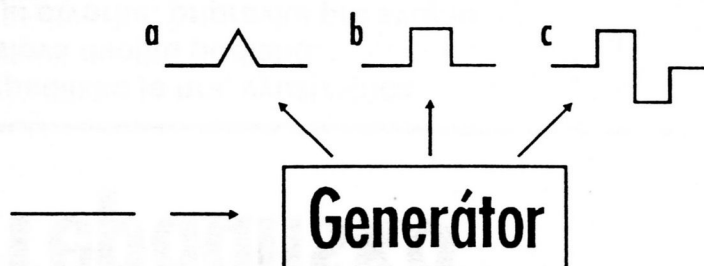
Napriek obrovskej rozmanitosti majú fraktály mnoho spoločných štruktúr a fraktálna grafika sa dá rozoznať na prvý pohľad. Všetko záleží od toho, s akým priblížením si necháme vyrábať výsledný obrázok. Čím detailnejšie ideme do hĺbky, tým väčšia rozmanitosť štruktúry sa nám odhaľuje pred očami. Začneme, samozrejme, od toho najjednoduchšieho princípu, ktorým je dnes už "klasická" fraktálna grafika Kochovej krivky (schematicky znázornená na obr. 1). V prvom priblížení fraktál vytvára pomerne jednoduchý útvar, ale pri opakovaní algoritmu s rovnakým vzorcom zachádzame do väčších a väčších detailov. Počiatočnou podmienkou je vytvorenie priamky. Do nej umiestnime zárez, vytvárajúci štyri rovnako dlhé úsečky. Tým sme vytvorili prvý krok iterácie riešenia. Útvar, ktorým realizujeme iteratívne kroky, sa nazýva generátor (obr. 2). Opakovaná aplikácia generátora nám vytvára charakteristickú jemnú štruktúru fraktálu. Jej rozlíšenie závisí už len na čase, presnosti a veľkosti dostupnej operačnej pamäti a je teoreticky neobmedzené.

"Kochov recept" určuje charakteristickú vzájomnú podobnosť fraktálov. Celý algoritmus riešenia jednotlivého kroku iterácie sa realizuje na 1/3 dĺžky počiatočnej úsečky. A ako to všetko uvidíme na

Obr. 1



Obr. 2



obrazovke? Fraktály Kochových kriviek vykresľujeme pomocou tzv. "korytnačej grafiky" (turtle-graphic). Vyzerá to, akoby čiary na obrazovke kreslila neviditeľná korytnačka, ktorá sa môže priamočiaro pohybovať vpred o určitú dráhu, prípadne sa otočí o určitý uhol. Jej stopy sú viditeľné čiary.

Výpis programu 1 obsahuje podprogramy nevyhnutné na ovládanie našej korytnačky. SET x,y dáva počiatočné súradnice postavenia korytnačky. TURN uhol ju otáča o zadaný uhol doľava a

nej krivky, pričom počiatočné súradnice sú dané príkazom SET xStart,yStart. Celé vykreslenie sa deje pomocou subrutiny Kresli. Tento podprogram využíva modernú metódu rekurzívneho volania, čo znamená, že podprogram si dokáže volať sám seba. Celý systém je pre názornosť uvedený v listingu 3. Ak má rad presnosti zo základného programu hodnotu $n=1$, potom parameter l má hodnotu 640, pričom sa vykoná v programe jeden

```
i(sp)=i(sp)+1
WEND
```

realizovať rovnakú úlohu pomocou cyklu FOR...NEXT:

```
FOR i(sp)=1 to gn
...
NEXT i(sp)
```

v ďalšom si vysvetlíme, prečo je potreb-

```
SUB MOVE (l) STATIC
  SHARED Smer,tx,ty,Farba
  rad=Smer/180*3.141593
  x=tx+l*COS(rad)
  y=ty+l*SIN(rad)/2
  LINE (tx,ty)-(x,y),Farba
  tx=x:ty=y
END SUB

SUB TURN (Uhol) STATIC
  SHARED Smer
  Smer=(Smer+Uhol) MOD 360
END SUB

SUB SET(x,y) STATIC
  SHARED tx,ty
  tx=x:ty=y
END SUB
```

Listing 1.

MOVE kroky pohybuje naším neviditeľným kresličom dopredu alebo dozadu (pri záporných hodnotách). Ak máme obrazovku nastavenú na rozlíšenie 320x256 alebo 640x512 bodov, nemusíme jednotlivé kroky pohybu transformovať. Pri rozlíšení 640x256 bodov je potrebné rozdiel pohybu na súradnici 'Y' vydeliť číslom 2. Zadávanie uhla otočenia je relatívne. Pri príkaze TURN 90 otočíme hlavu našej korytnačky hore. Ak príkaz zopakujeme, pohne sa zvieratko pri nasledovnom príkaze MOVE doľava. Výkričník za parametrom dáva číslo typu REAL s jednoduchou presnosťou.

Program "Koch1" z výpisu 2 používa rutinu Turtle. Jednotlivé podprogramy si môžete ľubovoľným spôsobom modifikovať a kombinovať. Program "Koch1" vytvára postupne čiary, tak ako sú uvedené na obr. 1. Jednoduchší variant vytvára akoby cestičku pre našu korytnačku v nasledujúcej verzii. Poľom g() sú popísané parametre uhlu otočenia. Počet opakovaní n v cykle FOR...NEXT udáva stupeň presnosti výpočtu kresle-

```
l=640:n=6:Farba=1
DIM i(n):gn=4

g(1)=0 :g(2)=60
g(3)=-120 :g(4)=60

xStart=0 :yStart=140

SET xStart,yStart :MOVE l

FOR Poradie=1 TO n
  LOCATE 22,40:PRINT "Dalej stlačením ľubovolnej klavesy!"
  WHILE INKEY$="" :WEND:CLS
  LOCATE 2,14:PRINT "Toto je krok iterácie";Poradie
  SET xStart,yStart
  sp=0:GOSUB Kresli
NEXT Poradie
LOCATE 22,40:PRINT "Koniec stlačením ľubovolnej klavesy!"
WHILE INKEY$="" :WEND
END
```

Listing 2.

```
Kresli:
  sp=sp+1:l=l/3
  i(sp)=1
  WHILE i(sp)<=gn
    TURN g(i(sp))
    IF sp<Poradie THEN
      GOSUB Kresli
    ELSE
      MOVE l
    END IF
    i(sp)=i(sp)+1
  WEND
  l=l*3:sp=sp-1
  RETURN
```

cyklus (prvé priblíženie), ktoré delí hodnotu l tromi a iba nám natočí smer korytnačky. Ak je $n=2$, príkazy za ELSE v listingu 3 sa vykonajú, program si načítava dáta z poľa g(), korytnačka sa pohybuje a kreslí fraktálnu líniu. Celý rad presnosti je daný vyšším počtom opakovaní cyklu. Ako to všetko prebieha sa najlepšie presvedčíte sami, keď si postupne budete zvyšovať počet opakovaní cyklu pre parameter $n>2$. Zistíte vlastne potrebný stupeň iterácie, keď opakované volanie podprogramu prestane fungovať, pretože rutiny sú zacyklené.

Poďme naspäť k metóde podľa výpisu 2. Skúsme namiesto sekvencie príkazov:

```
i(sp)=1
WHILE i(sp)<=gn
...
```

```
Kresli:
  l=l/3
  FOR i=1 TO gn
    TURN g(i)
    IF Poradie=2 THEN
      l=l/3
      FOR j=1 TO gn
        TURN g(j)
        IF Poradie=3 THEN
          l=l/3
          FOR k=1 TO gn
            TURN g(k)
            MOVE l
          NEXT k
        ELSE
          MOVE l
        END IF
      NEXT j
      l=l*3
    ELSE
      MOVE l
    END IF
  NEXT i
  RETURN
```

Listing 3.

né použiť príkaz WHILE... WEND.

Skúsme vychádzať rovnako ako v predošlej úvahe pri kreslení výpisom 3, zadáme stupeň presnosti $n=1$. Na rozdiel od dlhšej verzie kreslenia, zvýši program najprv parameter sp o 1. Pred volaním programu je jeho hodnota rovná nule, takto bude 1. Parameter sp nie je menší ako stupeň presnosti, príkazy za ELSE pohybujú korytnačkou. Zatiaľ nie je žiaden rozdiel oproti verzii z listingu

```

n=6:Farba=1
DIM i(n):gn=4

g(1)=0 :g(2)=60
g(3)=-120 :g(4)=60

Symetria=120
Pocet=360\Symetria
l=600/Pocet

xStart=200 :yStart=50

SET xStart,yStart :MOVE l

```

```

FOR Poradie=1 TO n
  LOCATE 22,40:PRINT "Dalej stlacenim lubovolnej klavesy!"
  WHILE INKEY$="":WEND:CLS
  LOCATE 2,4:PRINT "Toto je krok iteracie";Poradie
  SET xStart,yStart
  FOR i=1 TO Pocet
    sp=0:GOSUB Kresli
    TURN -Symetria
  NEXT i
NEXT Poradie
LOCATE 22,40:PRINT "Koniec stlacenim lubovolnej klavesy!"
WHILE INKEY$="":WEND
END
Kresli:
  sp=sp+1:l=l/3
  i(sp)=1
  WHILE i(sp)<=gn
    TURN g(i(sp))
    IF sp<Poradie THEN
      GOSUB Kresli
    ELSE
      MOVE l
    END IF
    i(sp)=i(sp)+1
  WEND
  l=l*3:sp=sp-1
RETURN

```

Listing 4.

```

gn=5

g(1)=0 :g(2)=90
g(3)=-90 :g(4)=-90 :g(5)=90

Symetria=90

xStart=310-1/3 :yStart=1/3

```

Listing 5.

```

gn=7:n=4

g(0)=0 :g(1)=90
g(2)=-90 :g(3)=-90
g(4)=0 :g(5)=90
g(6)=90 :g(7)=-90

Symetria=360

xStart=310-1/2:yStart=1/6+10

sp=sp+1:l=l/4

l=l*4:sp=sp-1

```

Listing 6.

3. V ďalšom kroku sa rad presnosti zvýši na hodnotu 2. Podmienka 'sp < rad presnosti' je pravdivá - subrutina Kresli sa - vyvolá. Cyklus s krokom i má hodnotu 2 t.j. i(2). Celá hra sa začína od začiatku - korytnačka sa štyri razy pootočí a vykreslí líniu. Pred opätovným zbehnutím cyklu zmenšíme hodnotu sp o 1. Celý priebeh cyklu realizujeme štyrikrát (FOR i(1)=1 TO 4), čím vznikne reťazenie typu:

```

FOR i(1)=1 TO 4
  FOR i(2)=1 TO 4
    ... a ďalej podľa požadovaného
    stupňa iterácie
  NEXT i(2)
NEXT i(1)

```

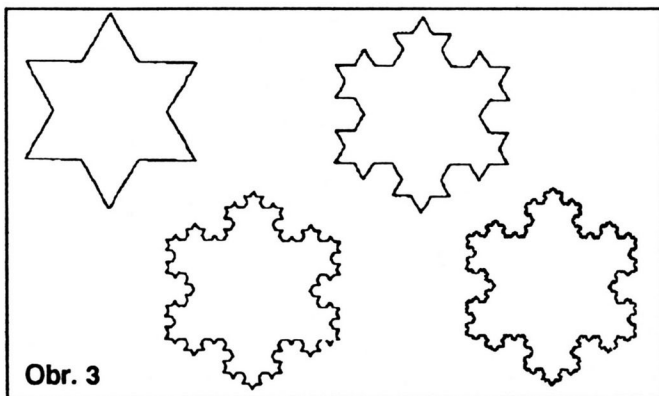
Výhody rekurzívnej metódy sú zrejmé. Programy sú kratšie a možno ich reťaziť. Jediné obmedzenie je presnosť iterácie, to znamená, do akého stupňa riešenia sa chceme dostať, pričom treba mať na

pamäti počet cyklov, o ktoré sa musíme vracat' späť. Všetky nezávislé premenné jednotlivých stupňov iterácie ukladáme do poľa, ktorého veľkosť je daná počtom reťazení cyklu. A tu prichádzame k problému náhrady slučky FOR...NEXT s cyklom WHILE...END. Je to výslovne len záležitosť použitého programového vybavenia, teda v našom prípade Amiga-Basic, ktorý neumožňuje deklarovať prvky poľa ako číselné premenné. Rekurzívne programovanie poskytuje vskutku vynikajúce možnosti.

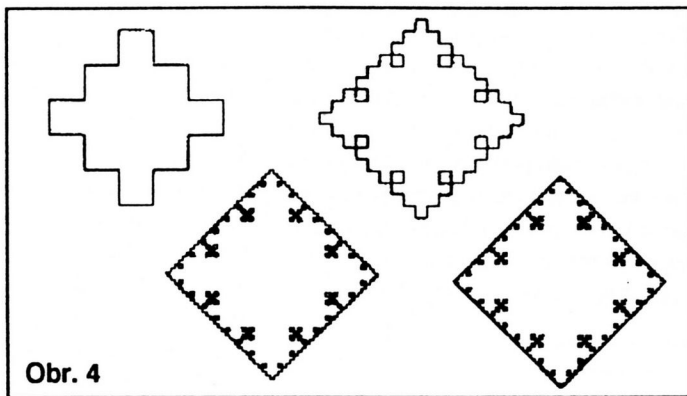
Ale poďme späť k fraktálnej grafike. Použitý algoritmus riešenia, t.j. Kochovej krivky znázorňujú akési pravidelné útvary v tvare snehových vločiek, a preto sa im niekedy hovorí aj "grafika snehových vločiek". Pomocou listingu 2 vytvoríme uzavretý cyklus, ak sa korytnačka otočí po každom pohybe o 120 stupňov a vznikne uzavretý útvar (obr. 3). Zmeňme

počiatočné podmienky a uvažujme o 360-stupňovej symetrii (90 stupňov dáva základný symetrický obrazec kvadrantu). Skúste vo výpise 4 experimentovať s uhlom a dĺžkami, výsledkom budú zaujímavé efekty. Pri použití generátorov iného typu (so silnejším delením) môžeme vytvoriť fraktály vyšších rozmerov (D). Generátor b na obr. 2 vytvára rozlíšenie fraktálu s rozmerom $D=1,46$. S rastúcim rozmerom však prístupuje problém prekríženia línií korytnačej grafiky. Už pri $D=2$ sa jednotlivé línie dotýkajú. Na štvorcovej Kochovej krivke ($D=1,46$) sa už línie križia. Aby ste ju vygenerovali, nahraďte príslušné riadky z výpisu 4 výpisom 5. Použitím generátora c z obrázku 2 dostaneme línie bez prekrižovania (obr. 5). Krivka má $D=1,5$ a je "strapatejšia". Pre jej modifikáciu je podstatná zmena parametrov podľa výpisu 6 pre program z výpisu 4. Tu sa dá experimentovať do omrzenia podľa chuti a času. Skúste si sami, ktorý rozmer fraktálu je ešte možný bez prekrižovania línií.

Kochova fraktálna grafika uvedeného typu má však svoje hranice a pravdepodobne by nás experimentovanie na tejto úrovni veľmi rýchlo omrzelo. Poďme teda ďalej a predstavme vzor, ktorému sa nedá jednoznačne definovať rozmer, resp. ktorý má rôzny rozmer v závislosti od oblasti fraktálu, v ktorej sa nachádza. Poďme si vypestovať fraktálny stromček. Ako na to? Skúsme neuzavrieť naše výpočty do útvaru, ale vychádzajme z jedného kmeňa s postupným rozvetvovaním do strán. Budeme pozorovať, ako sa naša korytnačka prehrýza cez jednotlivé halúzky kresleného stromu (s použitím výpisu programu 7). Spomaľme teraz kreslenie cyklu tým, že v listingu dáme poznámku v podprograme Kresli ako aktívny riadok, odstránime aj príkaz CLS v hlavnom cykle programu a pred príkaz WHILE INKEY\$ vložíme poznámku REM. V tomto programe už máme možnosť experimentovať s farbami fraktálu. Teraz môžeme ďalej fantazírovať. Čaká nás prominentná Mandelbrotova grafika plná farieb so zvláštnou čipkovou krajinou. Ocitli sme sa v rovine komplexných čísel. Komu matematika priveľmi nevoní, nemusí sa teraz obávať, že bude zahrnutý komplikovanou a nudnou teoriou čísel v komplexnej rovine. Po-



Obr. 3



Obr. 4

píšeme si len to najvyhnutnejšie, aby sme pochopili ten najzákladnejší základ.

Komplexné čísla sú zobrazenia dvojrozmerných hodnôt, preto pozostávajú z dvoch častí. Nami používaný basic umožňuje deklarovať iba jednej hodnoty pre premennú (ak ju nezadefinujeme ako pole), preto budeme používať dve premenné pre jedno komplexné číslo. Napríklad:

```
zr a zi
cr a ci
```

Prvý znak je názov premennej, druhý znak určuje reálnu (r) a imaginárnu (i) zložku komplexného čísla. Celková hodnota komplexného čísla je vyjadrená súčtom reálnej a imaginárnej zložky, teda:

$$z = zr + zi$$

Na zobrazenie Mandelbrotovej grafiky potrebujeme prakticky vykonať len jeden výpočet:

$$x = z^2 - c$$

x , z a c sú komplexné čísla. Základné operácie sčítania a odčítania sa vykonávajú ako súčet alebo rozdiel reálnej a imaginárnej zložky. V basicu to môže vyzeráť nasledovne:

```
xr = ar + br
xi = ai + bi
```

Násobenie komplexného čísla už nie je také jednoduché. Napríklad súčin $x = a \cdot b$ sa v Basicu vypočíta:

```
xr = (ar * br) - (ai * bi)
xi = (ar * bi) + (ai * br)
```

Z toho vyplýva nasledujúci vzorec pre druhú mocninu komplexného čísla:

```
xr = (zr * zr) - (zi * zi)
xi = 2 * (zr * zi)
```

K samotnej Mandelbrotovej množine sa dostaneme analýzou hodnôt postupnosti komplexných čísel

$$z \leftarrow z^2 - c$$

kde treba vhodne zvoliť veľkosť kroku iterácie podľa parametra c . Budeme vy-

chádzať z počiatočnej hodnoty $z = (0,0)$, pričom postupnosť bude závisieť na veľkosti parametra c , ktorý je veličinou závislou na počiatočných podmienkach. V tabuľke 1 sú hodnoty parametra c pri počiatočných podmienkach (1,1). V tomto prípade ak prekročí hodnota $r^2 + i^2$ veľkosť 2, postupnosť rastie cez všetky medze do nekonečna. Definícia Mandelbrotovej množiny pozostáva z komplexných čísel, pričom riešenie postupnosti zostáva v limitovaných hraniciach. Uvedený prípad s počiatočnými podmienkami v bode (1,1) nepatrí do definovanej množiny.

Po nevyhnutnom matematickom úvode si teraz ukážeme, ako to všetko dostaneme na obrazovku monitora. V každom kroku cyklu iterácie je nevyhnutné preskúšať, či je komplexný súčet $(r^2 + i^2) > 4$. Ušetríme si tým veľa času. V našom prípade za počiatočných podmienok $c = (1,1)$ je už po treťom kroku zjavné, že postupnosť bude rásť cez všetky medze. Tu prichádzame k určitej kontraverzii v programe. Potrebujeme nájsť optimálnu úroveň iterácie, aby sme nemuseli čakať do nekonečna na výpočet a aby bol obraz vypočítaný v dostatočnej presnosti. Počet krokov cyklu nám definuje tzv. hĺbku výpočtu, to znamená akú presnosť bude mať výsledný obrázok. Hĺbka výpočtu 40 dáva už veľmi prijateľné výsledky. Opäť tu máme priestor na experimentovanie, treba si len uvedomiť, že pekná vec niečo stojí - aj keď v našom prípade sa jedná o počítačový čas. Najkrajšie obrázky Mandelbrotovej grafiky dostaneme, keď sú interpretované farebne.

Ponúkame rýchly a efektívny algoritmus: Farba = Kroky modulo PočetFarieb

Počet krokov znamená vlastne programové splnenie nerovnosti - hodnota komplexného čísla > 2 . Ten, kto je pri experimentovaní netrpezlivý, môže si pomôcť malým trikom a používať malé obrazovky (zmenšiť hodnoty pre x_{max}

y_{max}), alebo program jednoducho nechať počítať a medzitým si vybaviť iné záležitosti. Pre skúsenejších programátorov doporučujeme celý výpočet zrealizovať v rýchlejšom programovacom jazyku C. Mandelbrotova množina je komplementárna so svojimi podmnožinami. Otvára sa pred nami čarovný svet farieb a tvarov, čím viac ideme do vnútornej štruktúry. Program z výpisu 8 dáva Mandelbrotovu grafiku. Aby sa množina vykreslila, treba riadok PSET... zameniť príkazom

```
IF it=itmax
THEN PSET(stpec,riadok),2
```

Program čaká na ukončenie stlačením ľubovoľnej klávesy. Ak chceme vykreslenú grafiku uložiť do pamäte, tak v basicu môžeme použiť štandardne dodávané programové vybavenie na diskete Extras 1.3 - program 'SaveACBM', ktorý je potrebné naviäzať na uvedené programové ukážky. Dostupné sú i programy bežiacie v pozadí, ktoré kombináciou určitých klúčov zosnímajú grafiku na obrazovke do operačnej pamäte, prípadne na disketu (Grabbit, Btsnap, Snip IFF, apod.) Mandelbrotova množina samozrejme nie je konečným riešením fraktálnej grafiky. Pre zaujímavosť sme na koniec pripravili jednu aplikáciu vo výpise 9 a to Juliovu množinu. Jej algoritmus je viazaný na rovnakú iteráciu komplexných čísel ako Mandelbrotova množina. Všetko sa deje určením parametra c . Ak sa Juliova množina nachádza v oblasti Mandelbrotovej množiny, je súvislá. Ak leží mimo hraníc, dostaneme izolované oblasti fraktálnej grafiky. Z matematického hľadiska môžeme označiť Mandelbrotove množiny ako zjednotenie všetkých komplexných parametrov c , kde príslušné Juliove množiny majú súvislý priebeh. Uvedený výpis je dobrým základom pre vlastné experimentovanie. Už len mať čas - a výsledok bude bohatou odmenou.

(AMIGA, 12/89, Spracované)

Listing 7.

```

l=180:n=9:Farba=1:PocetFarieb=4

xStart=300 :yStart=130 :ymax=WINDOW(3)

FOR i=0 TO 2
  FOR Poradie=1 TO n-i
    LOCATE 22,40:PRINT "Dalej stlacenim lubovolnej klavesy!"
    WHILE INKEY$="" :WEND:CLS
    SET xStart,yStart+65:Smer=90:MOVE 130!
    LOCATE 2,14:PRINT "Toto je krok iteracie";Poradie
    sp=0:GOSUB Kresli
  NEXT Poradie
  l=l*.6
NEXT i
LOCATE 22,40:PRINT "Koniec stlacenim lubovolnej klavesy!"
WHILE INKEY$="" :WEND
END

Kresli:
  REM for cakaj=1 to 1000:next cakaj
  sp=sp+1:l=l*.6:Farba=sp MOD PocetFarieb
  TURN 45! :MOVE 1
  IF sp<Poradie THEN GOSUB Kresli
  MOVE -1
  TURN -90! : MOVE 1
  IF sp<Poradie THEN GOSUB Kresli
  MOVE -1
  TURN 45!
  l=l/.6 :sp=sp-1: Farba=sp MOD PocetFarieb
RETURN

```

```

Hlbka=5
Modus=1

Sirka=(2*Modus MOD 2)*320
Vyska=(1+INT(Modus/2.1))*256
PocetFarieb=2^Hlbka
SCREEN 2,Sirka,Vyska,Hlbka,Modus
WINDOW 2,"Mandelbrot - grafika",,15,2
PALETTE 0,0,0,.5

ON BREAK GOSUB Koniec

xmax=WINDOW(2)
ymax=WINDOW(3)

rmin=-1 :rmax=2
imin=-1.5:imax=1.5
itmax=34

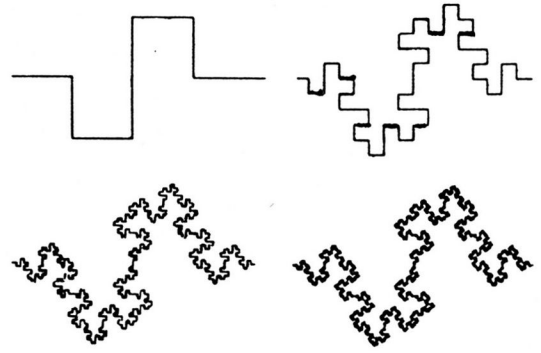
cr=rmin :ci=imax
dx=(rmax-rmin)/xmax
dy=(imax-imin)/ymax

FOR Riadok=0 TO ymax-1
  FOR stlpec=0 TO xmax-1
    it=0:zr=0:zi=0
    WHILE it<itmax AND (zr*zr+zi*zi)<4
      i=zi:r=zr
      zi=2*i*r-ci
      zr=r*r-i*i-cr
      it=it+1
    WEND
    Farba=it MOD PocetFarieb
    PSET (stlpec,Riadok),Farba
    cr=cr+dx
  NEXT stlpec
  cr=rmin:ci=ci-dy
NEXT Riadok

Koniec:
  WHILE INKEY$="" :WEND
  WINDOW CLOSE 2
  SCREEN CLOSE 2
END

```

Listing 8.



Obr. 5

```

Hlbka=5
Modus=1

Sirka=(2*Modus MOD 2)*320
Vyska=(1+INT(Modus/2.1))*256
PocetFarieb=2^Hlbka
SCREEN 2,Sirka,Vyska,Hlbka,Modus
WINDOW 2,"Julianovska mnozina - grafika",,15,2
PALETTE 0,0,0,.5

```

ON BREAK GOSUB Koniec

```

xmax=WINDOW(2)
ymax=WINDOW(3)

```

```

rmin=-2 :rmax=2
imin=-2 :imax=2

```

```
itmax=34
```

```

cr=.8 :ci=.8
zrs=rmin:zis=imax

```

```

dx=(rmax-rmin)/xmax
dy=(imax-imin)/ymax

```

```

FOR zeile=0 TO ymax-1
  FOR spalte=0 TO xmax-1
    it=0:zr=zrs:zi=zis
    WHILE it<itmax AND (zr*zr+zi*zi)<4
      i=zi:r=zr
      zi=2*i*r-ci
      zr=r*r-i*i-cr
      it=it+1
    WEND
    Farba=it MOD PocetFarieb
    PSET (spalte,zeile),Farba
    zrs=zrs+dx
  NEXT spalte
  zrs=rmin:zis=zis-dy
NEXT zeile

```

```

Koniec:
  WHILE INKEY$="" :WEND
  WINDOW CLOSE 2
  SCREEN CLOSE 2
END

```

Listing 9.

DiskEti

Dokončenie programu DiskEti z marcového čísla

```

271 JC4      END IF
272 KD2      END IF
273 L9       IF SpravaTyp=512& THEN
274 vH4       odpoved=AutoRequest (0&,FrageTxt,JaTxt,
NeinTxt,0&,0&,300,50)
275 X8       IF odpoved>0 THEN GOTO Koniec
276 OH2      END IF
277 pR0      RETURN
278 HI       PrintOut:
279 IG2      IF Tlac% = 0 THEN
280 O14      lo% = 0
281 O1       COLOR 3,0
282 mI       LOCATE 1,15
283 Xi       v=INT((50-LEN(Tlac$(Tlac%)))/2)
284 wN       PRINT SPC(v);Tlac$(Tlac%);
285 Me       COLOR 1,0
286 Dw2      ELSE
287 JA4      IF lo% = 1 THEN
288 tz6      LOCATE lo1%,42
289 Gz4      ELSE
290 x36      LOCATE lo1%,24
291 dW4      END IF
292 p1       v=INT((15-LEN(Tlac$(Tlac%)))/2)
293 PD       n=INT((15-LEN(Tlac$(Tlac%)))/2+.5)
294 qT       PRINT SPC(v);Tlac$(Tlac%);SPC(n);CHR$(32);
295 vs       lo% = lo% + 1
296 WO       IF lo% = 2 THEN
297 HI6      lo% = 0
298 mB      lo1% = lo1% + 1
299 le4      END IF
300 mf2      END IF
301 Dp0      RETURN
302 Rq       NovyInit:
303 9u2      novy% = 0
304 3e       Tlac% = 0
305 CA       stupen% = 0
306 KU       nacitanie% = 1
307 qg       teraz% = 0
308 ji       i% = 0
309 El       stuzka% = 0
310 CU       lo1% = 3
311 uB       tlacenie% = 0
312 I5      GOSUB UkazEtiketu
313 P10      RETURN
314 sI       VstupPola:
315 ur2      CALL OnGadget(df0Feld,AdresaOkna,0&)
316 yw      CALL OnGadget(df1Feld,AdresaOkna,0&)
317 NU      CALL OffGadget(EditFeld,AdresaOkna,0&)
318 pz      CALL OffGadget(WeiterFeld,AdresaOkna,0&)
319 fB      CALL OffGadget(OkFeld,AdresaOkna,0&)
320 vF      CALL OffGadget(GoonFeld,AdresaOkna,0&)
321 Z1      CALL OffGadget(DiskFeld,AdresaOkna,0&)
322 Uv      CALL OffGadget(DruckFeld,AdresaOkna,0&)
323 wd      CALL RefreshGadgets(df1Feld,AdresaOkna,0&)
324 aC0      RETURN
325 8M      PolePrec:
326 do2      CALL OffGadget(df0Feld,AdresaOkna,0&)
327 iu      CALL OffGadget(df1Feld,AdresaOkna,0&)
328 cx      CALL OnGadget(EditFeld,AdresaOkna,0&)

```

```

329 AP      CALL OnGadget(WeiterFeld,AdresaOkna,0&)
330 uF      CALL OnGadget(OkFeld,AdresaOkna,0&)
331 7V      CALL OnGadget(GoonFeld,AdresaOkna,0&)
332 e9      CALL OnGadget(DiskFeld,AdresaOkna,0&)
333 74      CALL OnGadget(DruckFeld,AdresaOkna,0&)
334 Qc      Vypln 4,69,632,97
335 zv      CALL RefreshGadgets(DruckFeld,AdresaOkna,0&)
336 mO0      RETURN
337 az      UkazEtiketu:
338 062      CLS
339 dF      LINE(171,10)-(461,151),3,b
340 Dm      LINE(172,10)-(172,151),3
341 EZ      LINE(314,10)-(314,151),3
342 Mj      LINE(315,10)-(315,151),3
343 Lf      LINE(460,10)-(460,151),3
344 3R      PRINT "Tlacim : "
345 vX0      RETURN
346 Rb      subrutiny:
347 4S2      SUB RezervPamete(Puffer,Velkost) STATIC
348 PV      SHARED VypisPamete
349 TZ      Velkost=Velkost+8
350 2E      Puffer=AllocMem(Velkost,65538&)
351 h7      IF Puffer>0 THEN
352 7p4      POKEL Puffer,VypisPamete
353 rm      POKEL Puffer+4,Velkost
354 RO      VypisPamete=Puffer
355 6f      Puffer=Puffer+8
356 L42     ELSE
357 LS4      ERROR 7
358 ib2     END IF
359 pr      END SUB
360 DX0     '-----
361 EO2     SUB VolnaPamet STATIC
362 dj      SHARED VypisPamete
363 KX      uvolnenie:
364 VH4     IF VypisPamete>0 THEN
365 b66     Adresa=PEEK(VypisPamete)
366 3f      Velkost=PEEK(VypisPamete+4)
367 Qc      FreeMem VypisPamete,Velkost
368 2d      VypisPamete=Adresa
369 aS      GOTO uvolnenie
370 un4     END IF
371 132     END SUB
372 I20     '-----
373 yi2     SUB OknoDEF(bs, x%, y%, b%, h%, IDCMP, f
, Feld, T$) STATIC
374 bh      Velkost=48+LEN(T$)+1
375 ic      RezervPamete bs,Velkost
376 AR      POKEW bs,x%
377 Ba      POKEW bs+2,y%
378 HB      POKEW bs+4,b%
379 me      POKEW bs+6,h%
380 6q      POKEW bs+8,65535&
381 sH      POKEL bs+10,IDCMP
382 pw      POKEL bs+14,f
383 1z      POKEL bs+18,Feld
384 Kg      POKEL bs+26,bs+48
385 sG      POKEW bs+46,1
386 GT      FOR i%=1 TO LEN(T$)
387 Nu4      POKE bs+47+i%,ASC(MID$(T$,i%,1))
388 LQ2     NEXT
389 JL      END SUB
390 mW0     '-----
391 Rg2     SUB FeldDef(bs,nx,x%,y%,b%,h%,f%,a%,T$,i,
txt,si,n%) STATIC
392 Ib      RezervPamete bs,44&
393 53      POKEL bs,nx
394 br      POKEW bs+4,x%
395 r0      POKEW bs+6,y%
396 xb      POKEW bs+8,b%
397 Xz      POKEW bs+10,h%
398 Uw      POKEW bs+12,f%
399 Fe      POKEW bs+14,a%
400 Ui      POKEW bs+16,T%

```

```

401 Sg   POKEL bs+18,i
402 Rk   POKEL bs+26,txt
403 5S   POKEL bs+34,si
404 U2   POKEW bs+38,n%
405 Zb   END SUB
406 Lf0
-----
407 Qr2  SUB IntText(bs,c1%,d1%,x%,y%,T$,nx) STATIC
408 Dn   Velkost=20+LEN(T$)+1
409 GA   RezervPamete bs,Velkost
410 Ko   POKE bs,c1%
411 10   POKE bs+2,d1%
412 t9   POKEW bs+4,x%
413 9I   POKEW bs+6,y%
414 8K   POKEL bs+12,bs+20
415 2j   POKEL bs+16,nx
416 kx   FOR i%=1 TO LEN(T$)
417 oA4   POKE bs+19+i%,ASC(MID$(T$,i%,1))
418 pu2   NEXT
419 np   END SUB
420 w30
-----
421 Yi2  SUB MacheRand(bs,x%,y%,C%,b%,h%) STATIC
422 2P   RezervPamete bs,48%
423 vC   POKEW bs,x%
424 wL   POKEW bs+2,y%
425 R2   POKE bs+4,C%
426 6U   POKE bs+7,b%
427 fi   POKEL bs+8,bs+16
428 wt   FOR i%=0 TO 1
429 HF4   POKEW bs+22+i%*4,h%-1
430 ys   POKEW bs+24+i%*4,b%-1
431 e9   POKEW bs+32+i%*4,1
432 cc   POKEW bs+38+i%*4,h%-1
433 7q   POKEW bs+40+i%*4,b%-2
434 5A2  NEXT
435 35   END SUB
436 WGO
-----
437 an2  SUB Sort STATIC
438 R8   SHARED max%
439 iF   Vypln 6,11,631,36
440 GU   tlacl "Triedim ",9,11:tlacl dir.Meno$(0),
9,19
441 kX   tlacl "Prosim trochu strpenia...",9,26
442 MD   lauf%=1:zaciatok%=1:koniec1%=max%-1
443 6s   FOR modus%=0 TO 1
444 oH4   WHILE lauf%=1
445 6H6   lauf%=0
446 5s   FOR i%=zaciatok% TO koniec1%
447 rg8   IF UCASE$(dir.Meno$(i%))>UCASE$(dir.
Meno$(i%+1)) THEN
448 JLA   SWAP dir.Meno$(i%),dir.Meno$(i%+1)
449 dx   SWAP dir.Typ$(i%),dir.Typ$(i%+1)
450 EQ   lauf%=1
451 D68   END IF
452 yJ6   NEXT i%
453 fT   zaciatok%=zaciatok%+1:lauf%=0
454 nE   FOR i%=koniec1% TO zaciatok% STEP -1
455 zo8   IF UCASE$(dir.Meno$(i%))<UCASE$(dir.
Meno$(i%-1)) THEN
456 XbA   SWAP dir.Meno$(i%),dir.Meno$(i%-1)
457 x9   SWAP dir.Typ$(i%),dir.Typ$(i%-1)
458 MY   lauf%=1
459 LE8   END IF
460 6R6   NEXT i%
461 G8   koniec1%=koniec1%-1
462 th4   WEND
463 IS2   NEXT modus%
464 WY   END SUB
465 ry0
-----
466 n72  SUB CitajDisk(dir$,max%) STATIC
467 M3   SHARED chyba%
468 tO   lies%=-2

```

```

469 rj   dir0$=dir$+CHR$(0)
470 jw   Puffer=252
471 yo   Adresa=Lock(SADD(dir0$),lies%)
472 x6   nacitanie%=0
473 Ea   chyba%=0
474 b4   IF Adresa=0 THEN
475 Ip4   Vypln 6,11,631,36
476 ta   tlacl "Obsah directory uz neexistuje!",9,19
477 St   FOR i%=0 TO 2000:NEXT
478 T8   chyba%=500
479 s6   EXIT SUB
480 gZ2  END IF
481 vm   opt=2^16
482 JJ   info=AllocMem(Puffer,opt)
483 vP   IF info=0 THEN ERROR 7
484 VX   uspech%=Examine$(Adresa,info)
485 bX   IF uspech%=0 THEN
486 T04   Vypln 6,11,631,36
487 Ov   tlacl "Obsah directory nemoze byt
prezrety!",9,19
488 d4   FOR i%=0 TO 2000:NEXT
489 eJ   chyba%=500
490 3H   EXIT SUB
491 rk2  END IF
492 f6   WHILE chyba%<>232
493 Am4   dir.Meno=info+8
494 Hh   FOR slucka%=0 TO 29
495 Cd6   check%=PEEK(dir.Meno+slucka%)
496 Bu   IF check%<>0 THEN
497 LF8   check$=check$+CHR$(check%)
498 dM6   ELSE
499 d58   slucka%=29
500 Ot6   END IF
501 004   NEXT slucka%
502 yG   dir.Meno$(nacitanie%)=check$
503 ht   check$=""
504 dG   typ=PEEKL(info+120)
505 kG   IF typ<0 THEN
506 dK6   dir.Typ$(nacitanie%)="DATA"
507 2a4   ELSEIF nacitanie%=0 THEN
508 P36   dir.Typ$(nacitanie%)="AKT.DIR."
509 oX4   ELSE
510 3T6   dir.Typ$(nacitanie%)="DIR"
511 B44   END IF
512 q2   check$=""
513 An   uspech%=ExNext$(Adresa,info)
514 40   IF uspech%=0 THEN
515 Pm6   chyba%=IoErr%
516 SB   IF chyba%<>232 THEN
517 yV8   Vypln 6,11,631,36
518 gi   tlacl "Chyba v obsahu directory!",9,19
519 8Z   FOR i%=0 TO 2000:NEXT
520 9o   chyba%=500
521 Ym   EXIT SUB
522 MF6   END IF
523 214   ELSE
524 hB6   nacitanie%=nacitanie%+1
525 ln   IF nacitanie%>max% THEN
526 5R8   chyba%=0
527 RK6   END IF
528 SL4   END IF
529 ym2  WEND
530 f4   CALL FreeMem(info,Puffer)
531 RZ   CALL Unlock(Adresa)
532 j4   max%=nacitanie%
533 df   END SUB
534 mt0
-----
535 U02  SUB Tien(Text$,mode%) STATIC
536 rA   POKEW WINDOW(8)+58,9
537 80   textlen=LEN(Text$)
538 j9   x%=PEEKW(WINDOW(8)+36)
539 qJ   y%=PEEKW(WINDOW(8)+38)
540 Wp   COLOR 2,0

```

```

541 K2 CALL Move(WINDOW(8),x%+2,y%+2)
542 Fx CALL Text(WINDOW(8),SADD(Text$),textlen%)
543 OX CALL SetDrMd(WINDOW(8),0)
544 Xp COLOR 1,0
545 ei CALL Move(WINDOW(8),x%,y%)
546 J1 CALL Text(WINDOW(8),SADD(Text$),textlen%)
547 7N IF mode%=0 THEN
548 Cp4 PRINT
549 ng2 END IF
550 ak CALL SetDrMd(WINDOW(8),1)
551 vx END SUB
552 4B0 '-----
553 ng2 SUB tlac1(txt$,x%,y%) STATIC
554 7u SHARED OknoRP
555 zS txtlen%=LEN(txt$)
556 YV CALL Move(OknoRP,x%,y%+7)
557 1w CALL Text(OknoRP,SADD(txt$),txtlen%)
558 24 END SUB
559 B10 '-----
560 eX2 SUB VypIn(x1%,y1%,x2%,y2%) STATIC
561 E1 SHARED OknoRP
562 OP CALL SetAPen(OknoRP,0)
563 k7 RectFill OknoRP,x1%,y1%,x2%,y2%
564 4U CALL SetAPen(OknoRP,1)
565 9B END SUB
566 sm0 '-----
567 o72 SUB StrInfo(stringpuffer,buffer$) STATIC
568 jP SHARED EditVelkost
569 Gn max%=15
570 hS IF LEN(buffer$)>max% THEN buffer$=LEFT$(buffer$,
max%)
571 7j IF(max% AND 1) THEN max%=max%+1
572 68 EditVelkost=36+2*(max%+4)
573 XS stringpuffer=AllocMem(EditVelkost,65536)
574 QO IF stringpuffer=0 THEN ERROR 7
575 7W POKEL stringpuffer,stringpuffer+36
576 RI POKEL stringpuffer+10,16
577 j5 IF buffer$<>" THEN
578 Ww4 FOR i%=1 TO LEN(buffer$)
579 Xs6 POKE stringpuffer+35+i%,ASC(MID$(buffer$,
i%,1))
580 RW4 NEXT
581 JC2 END IF
582 QS END SUB
583 930 '-----
584 Po2 SUB CitajEdit(stringpuffer,ret$) STATIC
585 Og SHARED EditVelkost
586 S4 puffer.string=PEEKL(stringpuffer)
587 2p ret$="":i%=0:check%=1
588 nN WHILE check%=0
589 ts4 check%=(PEEK(puffer.string+i%))
590 a6 ret$=ret$+CHR$(check%)
591 Ya i%=i%+1
592 zn2 WEND
593 z1 ret$=LEFT$(ret$,LEN(ret$)-1)
594 x8 CALL FreeMem(stringpuffer,EditVelkost)
595 df END SUB
596 mt0 '-----
597 Td2 SUB Mys STATIC
598 RI dummy=MOUSE(0)
599 EJ WHILE MOUSE(0)=0
600 7v WEND
601 j1 END SUB
602 sz0 '-----
603 n12 SUB Etikett STATIC
604 2K SHARED DruckFrageTxt,JaTxt,NeinTxt,Tlac%
605 vE odpoved=AutoRequest(0%,DruckFrageTxt,JaTxt,
NeinTxt,0%,0%,300,50)

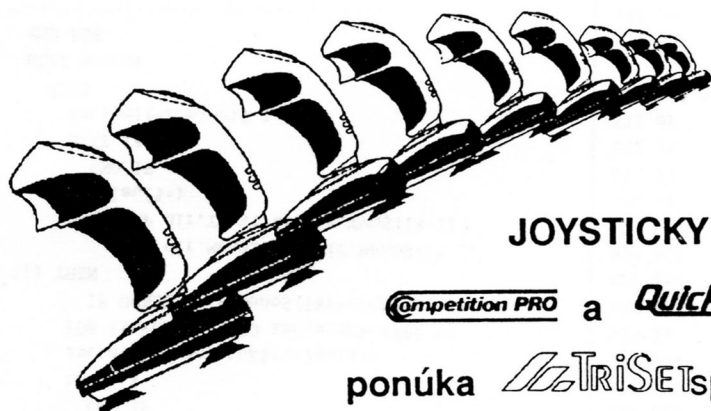
```

```

606 CM IF odpoved<1 THEN
607 sf4 GOTO prec
608 kd2 END IF
609 qO pocet%=18
610 eu esc$=CHR$(27):init$=esc$+"@"
611 FZ fett$=init$+esc$+"E"+esc$+"W"+CHR$(1)
612 Ja odstup$=esc$+"A"+CHR$(6)
613 bA maly$=esc$+"S0"
614 9B eng$=CHR$(15)
615 li elite$=esc$+"M"
616 Pt Ende$=esc$+"A"+CHR$(13)+esc$+"2"
617 As titul$=Tlac$(0)
618 X1 titul2$=LEFT$(Tlac$(0),13)
619 vb OPEN "par:" FOR OUTPUT AS #1
620 eN4 PRINT#1,init$;odstup$;
621 pd PRINT#1,maly$;eng$;
622 59 PRINT#1,SPC(12);"Ziadna ochrana proti
zapisaniu-->"
623 6F PRINT#1,
624 TM PRINT#1,SPC(19);"Ochrana proti
zapisaniu-->"
625 8H PRINT#1,
626 ZF einrueck=INT((48-LEN(titul$))/2)
627 sS justify=24-INT(LEN(titul$)/2+.5)
628 v9 PRINT#1,SPC(einrueck);titul$
629 tT einrueck=INT((14-LEN(titul2$))/2)
630 ZA PRINT#1,fett$
631 Ci PRINT#1,SPC(einrueck);titul2$
632 kB PRINT#1,esc$+"W0";
633 Ys PRINT#1,init$;
634 7A IF Tlac%>19 THEN
635 uO6 pocet%=32
636 Cl PRINT#1,odstup$;
637 1N PRINT#1,maly$;
638 E74 END IF
639 lb0 '--- Ramovanie ---
640 jn2 PRINT#1,elite$;CHR$(32);
641 oR FOR i=1 TO 15
642 fL4 PRINT#1,CHR$(45);
643 o42 NEXT i
644 KB PRINT#1,CHR$(32);
645 sV FOR i=1 TO 15
646 jP4 PRINT#1,CHR$(45);
647 s82 NEXT i
648 5V PRINT#1,CHR$(32)
649 se FOR i=1 TO pocet% STEP 2
650 MZ4 v1=INT((15-LEN(Tlac$(i)))/2)
651 hp n1=INT((15-LEN(Tlac$(i)))/2+.5)
652 IF v2=INT((15-LEN(Tlac$(i+1)))/2)
653 Ih n2=INT((15-LEN(Tlac$(i+1)))/2+.5)
654 I3 PRINT#1,CHR$(124);SPC(v1);Tlac$(i);
SPC(n1);CHR$(124);
655 2y PRINT#1,SPC(v2);Tlac$(i+1);SPC(n2);
CHR$(124)
656 1H2 NEXT i
657 XO PRINT#1,CHR$(32);
658 5i FOR i=1 TO 15
659 wc4 PRINT#1,CHR$(45);
660 5L2 NEXT i
661 bS PRINT#1,CHR$(32);
662 9m FOR i=1 TO 15
663 Og4 PRINT#1,CHR$(45);
664 9P2 NEXT i
665 Tp PRINT#1,maly$;
666 76 PRINT#1,Ende$
667 zm CLOSE #1
668 hp prec:
669 Pg FOR i%=0 TO 32
670 xc4 Tlac$(i%)=""
671 uz2 NEXT
672 su END SUB

```

(koniec programu - pozn. red.)



QuickShot
by Bandwell

JOYSTICKY od firiem

Competition PRO a **QuickShot**

ponúka **TriSet** spol. s r.o.

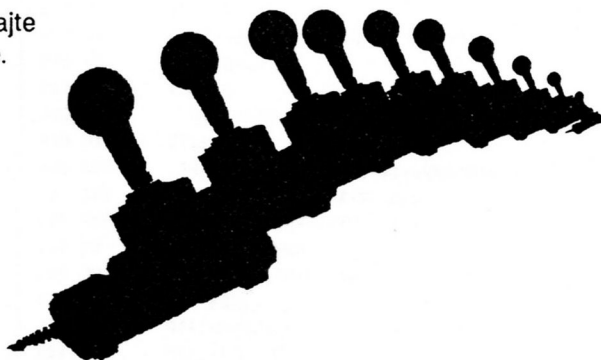
Competition Pro 5000 570,- Kčs (vrátane dane z obratu)

Competition Pro Star 670,- Kčs (vrátane dane z obratu)

Competition Pro Extra 770,- Kčs (vrátane dane z obratu)

Apache DeLuxe 266,- Kčs (vrátane dane z obratu)

Objednávky posielajte
na adresu redakcie.



Competition PRO

TRKA
press

POČÍTAČ
AKTÍVNE