

AMIGA

MAGAZYN

2/92 Miesięcznik fanów komputera Amiga

Rodzina Motoroli

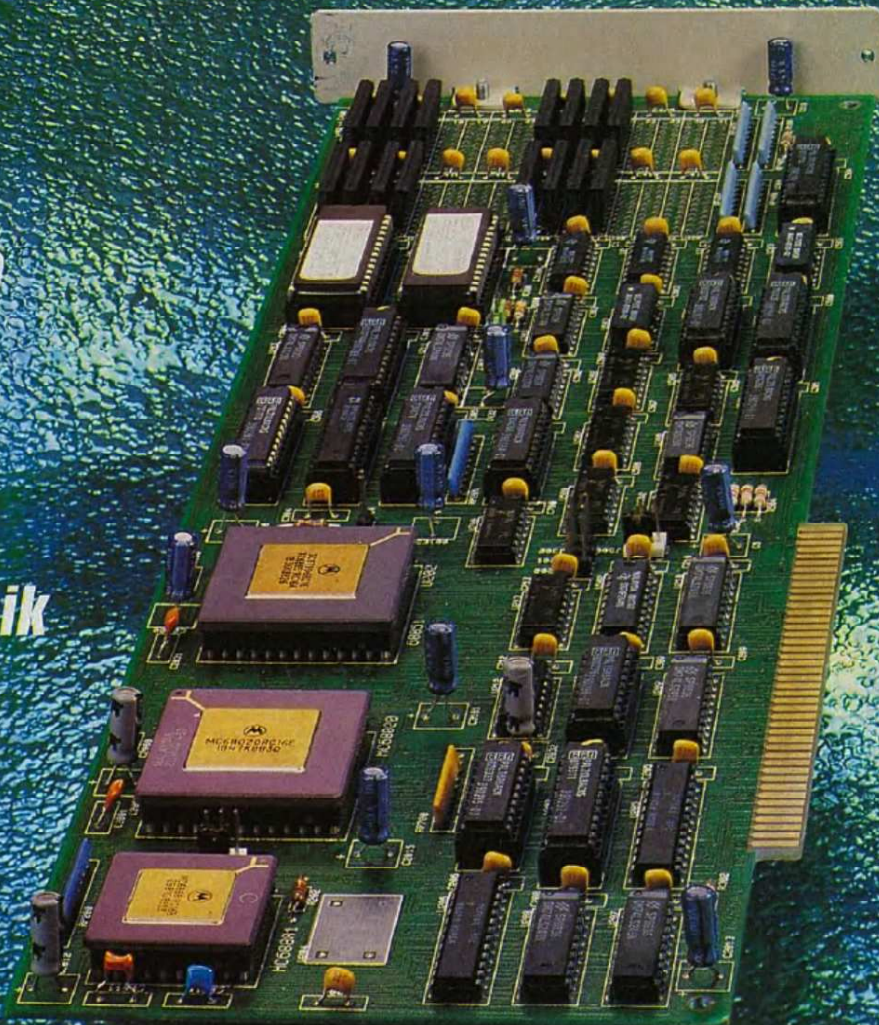
Karty Turbo

Animacja

MikroKino

Zęb to sam

**Przełącznik
drukarek**



AKTUALNOŚCI

Informacje 4

KARTY TURBO

Procesory rodziny 68000 6

Przetestujmy 10

Stoper 13

Karty 60000/68020 15

Podsumowanie 18

OD REDAKCJI

Zasady prenumeraty 21

ANIMACJA

Mikro-KINO 23

MIKROKINO

AMIGA
play

Ja w kwestii formalnej 59

Megatraveller II 60

Skarabeusz 61

Lure of Temtress 63

Zone Warrior 65

Nielegalny doping 66

Podpowiedzi do Civilization 68

Lista przebojów 69



ZACZYNAMY

I Ty to potrafisz 26

DIALOGI

Do łezki łezka 28

AMIGA PO POLSKU

Amiga a sprawa polska - standard xJP 32

Klawiatura z Alaski 34

KURS

Szlachectwo zobowiązuje (cz. 2) 36

Składnia rozkazów ARexxa 40

HYDE PARK

Hyde Park 47

PROGRAMY KOPIUJĄCE

Przegląd programów kopiujących (cz. 2) 48

GRAFIKA

Kolorowy Pitagoras 51

Rozwiązanie ćwiczeń z cz. 1 57

HARDWARE

Podwójne zabezpieczenie 70

TIPS & TRICKS

Kuferek 73

PUBLIC DOMAIN

Tanie i legalne 76

KONKURS

Wielki konkurs 78

LISTY

Odpowiedzi na listy 79

Magazyn AMIGA

Miesięcznik fanów
komputera Amiga
Numer drugi, rok pierwszy
Październik '92
Nakład: 45.000
PL ISSN 1230-1345,
nr indeksu 394933
(c) Copyright
by Wydawnictwo LUPUS
Amiga i Commodore są
nazwami zastrzeżonymi
firmy Commodore-Amiga
Tłumaczenia z miesięcznika
"AMIGA Magazin"
za zgodą wydawnictwa
Markt & Technik, RFN

Redaguje kolegium:

Wojciech Bruszewski
Piotr Drapich
Grzegorz Eider
(p.o. red. naczelnego)
Eiżbieta Kozakiewicz
(sekretarz redakcji)
Krzysztof Młynarski
Marek Pampuch
(z-ca red. naczelnego)
Rafał Wiosna
Stale współpracują:
Piotr Hebisz (Wrocław)
Janusz Konopka (Wrocław)
Robert Korzeniewski
(Warszawa)
Sambor Kuźma (Szczecin)
Jan Pikul (Mielec)

Adres redakcji:

PL 00-739 Warszawa
ul. Stępińska 22/30
tel. (0-22)410031 w. 128, 154
fax (0-22)410374
(10.00-16.00)
tlx 813527 omig pl

Layout na podstawie "AMIGA Magazin":

Wydawnictwo LUPUS
Piotr Kakiêt

DTP i skład komputerowy:

Artur Gąsiorek

Redakcja techniczna:

Jadwiga Pajewska

Fotografia okładkowa:

Bogusz Steinborn

Dział Reklamy:

Piotr Roszczyk

Reklamy przyjmowane
są w redakcji.

Za treść ogłoszeń redakcja
nie odpowiada.

Maszynopisów

nie zamówionych

redakcja nie zwraca.

Redakcja zastrzega sobie
prawo do wprowadzania
skrótów, własnych tytułów
i śródtytułów
w nadsyłanych materiałach.

MELODIA PRZYSZŁOŚCI

Tematem wiodącym tego numeru Magazynu AMIGA są karty przyspieszające działanie Amigi, popularnie zwane "turbokartami", a slangowo "dopalaczami". Dokładny test trzydziestu trzech takich kart, z których niektóre znajdują się dopiero w fazie prototypu — pozwoli przybliżyć Czytelnikom ten temat. Zdajemy sobie sprawę, że dla większości z nas zakup takiej karty jest marzeniem ściętej głowy. Taki "dopalacz" potrafi zmusić Amigę do tego, aby działania, które normalnie zabierają 10 godzin, zostały wykonane w niecałe 10 minut (to nie pomyłka — niektóre karty uzyskują wyniki czasowe 64 razy lepsze niż "zwykła" Amiga). Jednak koszt takiej karty prawie dziesięciokrotnie przewyższa i tak niemałą kwotę, jaką każdy z nas wydał na komputer. Nie należy jednak popadać w kompleksy. Dwa lata temu najtańsza karta z procesorem Motorola 68030 kosztowała 7500 marek, dziś — jeśli się dobrze poszuka i odejmie podatek MwSt. — można nabyć taką kartę prawie ośmiokrotnie taniej. A zatem — poczekamy, zobaczymy. Nie można jednak czekać bezczynnie. Warto czas oczekiwania wykorzystać choćby na zdobycie wiedzy o takich kartach i zastanowienie się, czy rzeczywiście jest to coś, "co tygrysy lubią najbardziej".

Rzecz jasna, także i ci Czytelnicy, których nie zainteresuje temat kart, znajdą w tym numerze coś dla siebie. Oprócz kontynuacji stałych działów i "seriali" w magazynie można także przeczytać artykuł, z którego dowiedziecie się, jak z Amigi zrobić IBM-a. Rozpoczyna on kolejny cykl, który będzie kontynuowany w następnych numerach.

Nie zapomnieliśmy także o obiecanej niespodziance. Rozpoczynamy nasz wielki KONKURS. Postanowiliśmy znaleźć najlepszego znawcę Amigi w Polsce. Aby na starcie nie odbierać nikomu szans — pytania mają charakter testowy. Spróbujcie. Wydaje nam się, że warto. A może akurat uda się narysować krzyżyk we właściwej kratce. Z drugiej strony, aby wyrobić w Was niezbędną każdemu komputerowcowi cechę zwaną cierpliwością, a ponadto sprawić zwycięzcom miły prezent pod choinkę — konkurs będzie składał się z trzech testów zamieszczonych w kolejnych trzech numerach. O szczegółach przeczytacie wewnątrz magazynu.

Marek Pampuch

Infinity RAM!

Amerykańska firma M.A.S.T. wprowadziła na rynek kartę rozszerzenia pamięci do Amigi 2000, umożliwiającą zwiększenie zasobów RAM tego komputera do 64 MB (sic!). Karta ta została nazwana Infinity Machine. Według producenta jest ona przeznaczona przede wszystkim dla użytkowników zajmujących się pracami z dziedziny multimedialnych.

Za 3 dolary...

Norweska firma Saarbrot Ltd. oferuje pióro świetlne przeznaczone do Amigi 3000 wraz z uniwersalnym driverem za jedyne 3 USD. Według materiałów reklamowych oferta ta należy do najtańszych w swej klasie.

Komputer i szkoła...

Firma Kosmos Software Ltd. z Wielkiej Brytanii jest jednym z największych dostawców oprogramowania edukacyjnego na tamtejszym rynku. Według najnowszych informacji, do oferty tej firmy należą programy pomagające dzieciom i młodzieży przyswoić sobie informacje m.in. z następujących dziedzin:

- języków: francuskiego, angielskiego, niemieckiego, hiszpańskiego i włoskiego;
- historii, geografii, nauki o nauce, wiadomości ogólnych, piłki nożnej, sportu, udzielania pierwszej pomocy, historii naturalnej, spellingu oraz arytmetyki.

Dodatkowo programy te są tak skonstruowane, że umożliwiają osobom nauczającym (rodzicom, nauczycielom), uzupełnianie materiału szkoleniowego we własnym zakresie. Szkoda tylko, że w naszym kraju nie ma takiego wyboru oprogramowania edukacyjnego.

Amiga i medycyna...

W USA grafika komputerowa Amiga została wykorzystana do celów medycznych. Przy użyciu specjalistycznego

oprogramowania, generującego na ekranie tzw. figury niemożliwe, bada się niektóre schorzenia układu nerwowego. Według uczonych, zajmujących się tymi badaniami, dynamiczna grafika komputerowa, połączona z dużą paletą barw, umożliwia wykonanie wielu badań, których przeprowadzenie metodami tradycyjnymi byłoby właściwie nierealne.

Hard drive i eresy!

Duńska firma Finn R. Jacobsen ApS wprowadziła na rynek tzw. HardCard, czyli kartę kontrolera SCSI z zainstalowanym twardym dyskiem o pojemności 105 MB, firmy Quantum. Nie byłoby w tym nic nadzwyczajnego, gdyby nie fakt, że na tej karcie znajdują się dwa dodatkowe porty szeregowo standardu RS-232C. Rozwiązanie takie umożliwia oprócz podłączenia urządzeń zgodnych ze standardem SCSI (twarde dyski, streamery i inne) także np. modemów czy drukarek. Karta ta jest sprzedawana w dwóch wersjach: z twardym dyskiem albo jako tzw. Host Adapter bez dysku. Według danych producenta, karty te współpracują zarówno z Amigami 2000, jak i 3000. Ta sama firma oferuje także streamer na kasety o pojemności 150 MB o nazwie DC600XTD wraz z niezbędnym oprogramowaniem.

Track Display

Niemiecka firma Alcomp oferuje niewielką przystawkę do komputerów Amiga, instalowaną w porcie zewnętrznych stacji dyskowych. Przystawka ta, o nazwie Track Display jest wyposażona w wyświetlacz LCD, pokazujący informacje o numerze aktualnie obsługiwanej ścieżki w jednym z czterech, możliwych do przyłączenia do Amigi napędów dyskietek (df0:, df1:, df2:, df3).

A może by tak ołówkiem...

Kanadyjska firma Anakin Research wprowadziła na rynek europejski produkt o nazwie Easy! Tegneplade. Jest to tablica graficzna o

rozmiarach 21 x 33 cm i rozdzielczości 1024 x 1024 punkty, umożliwiająca tworzenie na ekranie komputera dowolnych obrazków w wyjątkowo naturalny sposób, a mianowicie przy użyciu zwykłego ołówka. Jest to oczywiście możliwe dzięki specjalnemu oprogramowaniu dostarczanemu przez producenta wraz ze sprzętem. Zarówno sam Easy!, jak i oprogramowanie współpracują ze wszystkimi modelami Amig.

Joy-komputer!

W Kanadzie pojawiło się ostatnio bardzo dziwne urządzenie do Amigi. Na pierwszy rzut oka, wygląda jak zwykły dżojstik podłączony do komputera poprzez jakiś interfejs (pudełeczko z małym wyświetlaczem alfanumerycznym LCD). Urządzenie to nazywa się Mousestick i jest przeznaczone dla osób wykorzystujących oprogramowanie typu CAD/CAM. Istnieje między innymi możliwość programowania rozdzielczości tego manipulatora, która może być ustawiona na maksymalną wielkość równą 1200 dpi (l). Można też programować płynność reakcji na ruchy dżojstika oraz funkcje przycisków. Wszystko to staje się możliwe dzięki opisanemu wyżej małemu pudełeczku z wyświetlaczem, które zawiera programowalny kontroler z pamięcią i podtrzymaniem baterijnym.

Mouse Master!

Amerykańska firma Practical Solutions opracowała ciekawe urządzenie o nazwie Mouse Master. Jest to niewielkie szare pudełeczko z trzema wbudowanymi portami joystickowymi i dwoma wyprowadzonymi przewodami, zakończonymi takimi samymi wtykami, jakie spotykamy w każdym joysticku do Amigi. Oba te kable należy zainstalować w portach joystick/mouse Amigi. Do Mouse Mastera podłączamy oba joysticki i mysz, i już gotowe! Dzięki przetacznikowi znajdującemu się na wierzchu obudowy możemy teraz wybierać, czy chcemy korzystać z myszy, czy joysticka, bez

konieczności dokonywania zmiany urządzeń zainstalowanych w porcie Amigi. Rozwiązanie takie ma dwie podstawowe zalety: jest wygodne oraz oszczędza porty wbudowane w komputer.

Amiga i TV-text

Angielska firma Rendale Ltd. wyprodukowała genloc oznaczony jako A8702, umożliwiający oprócz standardowych funkcji takiego urządzenia, również odbiór telewizyjny i magazynów reklamowych rozpowszechnianych drogą telewizyjną (TV-Show). Aby umożliwić użytkownikom tej przystawki korzystanie z dodatkowych funkcji, firma Rendale Ltd. dołącza do zestawu specjalny program o nazwie ProVideo Plus.

EtherNet i Amiga...

Amerykańska firma Hydra (będąca resztą filii znanej GVP), wprowadziła na rynek kartę sieciową standardu Ethernet współpracującą z Amigami 2000 i 3000. Karta ta jest w pełni kompatybilna z Ethernet Version 2/IEEE 802.3 standard typu A i B. Dzięki zastosowaniu tej karty można korzystać z prędkości transmisji 10Mb/s. Karta wykorzystuje do swoich celów bezpośrednią transmisję przez dostępne w Amidze 16-bitowe kanały DMA. Dodatkowo na karcie znajduje się 16 KB pamięci typu SRAM, stanowiącej bufor dla DMA. Użycie tej karty umożliwia podłączenie Amigi do np. sieci typu Novell lub DECnet.

Krzysztof Młynarski

OCEAN walczy z piratami

Firma Ocean Software już od lat znana jest z niekończącej się wojny z piratami nielegalnie sprzedającymi jej oprogramowanie. W czasach ZX Spectrum wprowadziła "Speedlock" — system zabezpieczenia oprogramowania, który był niesamowicie trudny do złamania. Jednak wprowadzenie na rynek komputerów 16-bitowych pociągnęło



za sobą dwa ważne fakty. Po pierwsze dał się zauważyć bardzo znaczny wzrost liczby pirackich kopii na rynku, co powoduje straty finansowe. Po drugie stało się możliwe opracowanie takich systemów zabezpieczeń, dzięki którym złamanie oprogramowania jest bardzo, bardzo trudne, jeżeli nie niemożliwe — cena programów na "16-bitowce" jest na tyle wysoka, że pozwala na finansowanie prac nad takimi systemami zabezpieczenia.

Od dawna znane są tzw. dongle, czyli małe kartridże — moduły, bez których program nie działa (tak między innymi zabezpieczony jest popularny na rynku PC program Ventura Publisher). Ich cena w masowej produkcji jest bardzo mała, a urządzenia takie zmniejszają prawdopodobieństwo użytkownika (a nie wykonania) pirackiej kopii programu.

Podchwytując ten pomysł, Ocean Software zleciło angielskiej firmie o ciekawej nazwie Worx-4-Me opracowanie takiego dongla dla komputerów Atari ST, Amiga i PC. Problem polega na tym, aby tak zaprojektować to urządzenie, żeby maksymalnie utrudnić analizę i zmianę kodu programu — na przykład przez zapisanie w pamięci dongla pewnego wycinku kodu, bez którego oprogramowanie nie będzie działać.

Pierwszą grą, która wykorzystuje tę przystawkę jest Robocop 3, wypuszczony stosunkowo niedawno przez Ocean. Bossowie firmy marzą o tym, aby gra nie została "złamana" w okresie pierwszych — decydujących miesięcy sprzedaży.

Jednak życie jest o wiele brutalniejsze — zaprawieni w bojach łamacze kodu, popularni "crackerzy" szczęśliwie (dla nich) uporali się z zabezpieczeniem i piracka kopia Robocop 3 od dawna dostępna jest na przykład na warszawskiej giełdzie.

I ponownie, w wojnie dłuższej niż Druga Wojna Światowa, pozostaje nam czekać albo na cud, albo na coś, co powstrzyma przemyśl oprogramowania rozrywkowego przed upadkiem.

COPY PARTY W WARSZAWIE

W dniach od 6. do 8. listopada odbędzie się w Warszawie zjazd użytkowników komputera Amiga. Miejscem imprezy będzie szkoła nr 220 przy ulicy Jana Pawła II, róg Grzybowskiej (ale nie tam, gdzie mieści się popularna giełda, ale po przeciwnej stronie ulicy). Całość będzie trwała od piątku, od godziny 17:00, do niedzieli. Organizatorzy nie gwarantują noclegu, ale za to odbędą się wiele konkursów, m.in. na najlepszą grafikę, muzykę i program demonstracyjny, które przewiozą ze sobą największe polskie grupy — Joker, G-Force, Action Direct, Katharsis i wiele innych. Po więcej informacji oraz zaproszenie radzimy zwrócić się z pisemną prośbą na adres:

Robert Korzeniewski — Mr. Root, skr. poczt. 23, 05-807 Podkowa Leśna. Można też zatelefonować (022)-178883 w godzinach 9:00-15:30.

Rafał Wiosna

Amiga w Pakiecie

Commodore oferuje Amigi z różnymi dodatkami z zakresu Desktop-Video jako pakiety. Z programem Scala 500 pakiet nazywa się Video Titler, zaś ze Scalą 1.13 - Video-Studio. Wszystkie pakiety mają przystępne ceny.

Zestaw 1: Amiga 500 Plus, Genlock HG 1, Scala 500. Cena: ok. 500 DM.

Zestaw 2: CDTV, Genlock CD 1301, klawiatura, stacja dysków, myszka, Scala 500 na kompakt dysku. Cena: ok. 2400 DM.

Zestaw 3: Amiga 2000, monitor 1084 S, Genlock A 2301, Scala 500. Cena: ok. 2600 DM.

Zestaw 4: Amiga 2000, monitor 1084 S, RAM A 2058-2, kontroler SCSI A 2091 z twardym dyskiem 50 MB, Genlock A 2300 Y/C, Scala 1.13. Cena: ok. 4600 DM.

Zestaw 5: Amiga 2000, monitor 1084 S, pamięć RAM A 2058-2, kontroler SCSI A 2091 z twardym dyskiem 50 MB, Genlock DVE 10P, Scal 1.13. Cena: ok. 6400 DM.

Zestaw 6: Amiga 3000, twar-

dy dysk 50 MB, monitor 1084 S, 4 MB RAM-u, Genlock A 2301, Scala 1.13. Cena: ok. 6400 DM.

Zestaw 7: Amiga 3000, twarde dysk 50 MB, monitor 1084 S, 4 MB RAM-u, Genlock A 2300 Y/C, Scala 1.13. Cena: ok. 6700 DM.

Zestaw 8: Amiga 3000, twarde dysk 50 MB, monitor 1084 S, 4 MB RAM-u, Genlock DVE 10P, Scala 1.13. Cena: ok. 9000 DM.

Agfa Intellifonts

Skalowalne czcionki Agfa Intellifonts są już dostępne także na Amigę. Działają one z systemem operacyjnym w wersji 2.0. Ta wielostronna technologia liternictwa produkowana jest wspólnie przez Agfę i Hewlett-Packarda. Intellifonts zarówno na ekranie, jak i w wydruku we wszystkich kierunkach stopniowo zmniejsza lub zwiększa czcionki, gwarantując wysoką jakość wydruku także w niskiej rozdzielczości drukarki laserowej i igłowej. Propozycja Agfy obejmuje duży wybór różnych czcionek.

Tekstury

Profesjonalnym użytkownikom programów Ray-Tracing firma Werbedesign & Computerrafik Stang oferuje dużą liczbę nowych tekstur. Grafiki stworzone w 24-bitowym formacie nadają się do wszystkich Ray Tracerów z funkcją Textur/Brushmapping. Na dziesięciu dyskietkach znajduje się ogółem 10 obrazków o rozdzielczości 512 x 512 punktów. Dodatkowo na każdej dyskietce znajduje się tekstura w trybie HAM 256 x 256 punktów. Do nabycia są trzy wydania — każde z dziesięcioma teksturami z różnego zakresu jak: drewno, kamienie, materiały, korki, woda, marmur itd.

Drukarka DIN A2

PROTRACER

Protracer jest drukarką atramentową zdolną do wykonywania rysunków z CAD-a o formacie kartki do DIN A2. Drukuje ona monochromatycznie

rysunki z CAD-a, DTP i teksty w wydruku o jakości podobnej jak na drukarce laserowej. Protracer działa techniką Bubble-Jet na papierze normalnym, rysunkowym i na folii. Drukarka bazuje na mechanizmie Canon i szybkim procesorze Intel 1960-RISC. Na wydruk rysunku o formacie DIN A2 Protracer potrzebuje nie więcej niż pięć minut. Największą rozdzielczością drukarki jest 360 dpi, akceptowane są kartki większe od DIN A4 oraz DIN A3 i DIN A2. Drukarka może być rozbudowana o karty emulujące PostScript.

AMIGA Magazin 8/92
Tłum. Bartosz Pampuch

Nowa Amiga !!!

Na początku września odbyły się w Pasadenie (USA) targi komputerowe, na których firma Commodore przedstawiła swój najnowszy komputer — Amigę 4000.

Jest ona uważana za wielki technologiczny skok w historii Amigi. Dysponuje bardzo rozbudowaną grafiką i dźwiękiem — do znanych już z A3000 nowych rozdzielczości doszły następne (m.in. typowe rozdzielczości VGA i SVGA.) oraz zwiększyła się liczba kolorów. Zastosowano nowe kości graficzne oznaczone skrótem AGA (Advanced Graphics Architecture), pozwalające na wyświetlanie obrazów i animacji w każdej z dostępnych rozdzielczości w maksymalnie 256.000 kolorach z palety 16,8 milionów. Dźwięk jest czterokanałowy, ale za to zapisywany na 16 bitach (tak jak w odtwarzaczach CD). Komputer wyposażony jest w procesor Motorola 68040, 6 MB RAM (maksymalnie 64 MB), stację dysków HD oraz twarde dysk 120 MB.

Dzięki modularnej konstrukcji (procesor jest umieszczony nie na płycie głównej, ale na specjalnej płycie) można instalować nowe wersje procesora, gdy tylko pojawią się na rynku. Cena nowej Amigi, zalecana przez producenta to 3699 dolarów, co jak na taką konfigurację nie jest ceną wygórowaną.

Piotr Drapich

Peter Spring

Możliwości każdego modelu Amigi można znacznie powiększyć przez wspomnianą wyżej wymianę procesora lub przez zastosowanie karty turbo z wydajniejszym procesorem i wyższą częstotliwością taktowania zegara. Sercem większości Amig jest jednak stary poczciwy:

MC68000

Prawie wszystkie cechy MC68000 klasyfikują go do procesorów 16-bitowych, jednak jego architektura wewnętrzna przypomina procesory 32-bitowe. Niestety, ma on jedynie 24-bitową szynę adresową (a właściwie 23 oraz kombinacja /UDS i /LDS), potrafi też zaadresować tylko 16 MB pamięci. W przypadku takiej "hybrydy" od razu pojawia się pytanie o zgodność programową. Czy na Amidze wyposażonej w MC68000 będzie można używać na przykład najnowszego oprogramowania? W większości przypadków tak. Dzięki 32-bitowej architekturze poza nielicznymi wyjątkami nie ma przeszkód w pracy z oprogramowaniem dostosowanym do procesorów 32-bitowych. Inna sprawa, że nieraz może się to odbywać z prędkością żółwia (dlaczego — przeczytajcie dalej). Aby przyspieszyć nieco szybkość działania oprogramowania, można wymienić ten procesor na jego następców (68010, 68020, 68030, 68040), bowiem nie tylko struktura, ale i wszystkie rejestry adresowe, a także wskaźniki stosu i licznik programu (Program Counter) w MC68000 są również 32-bitowe.

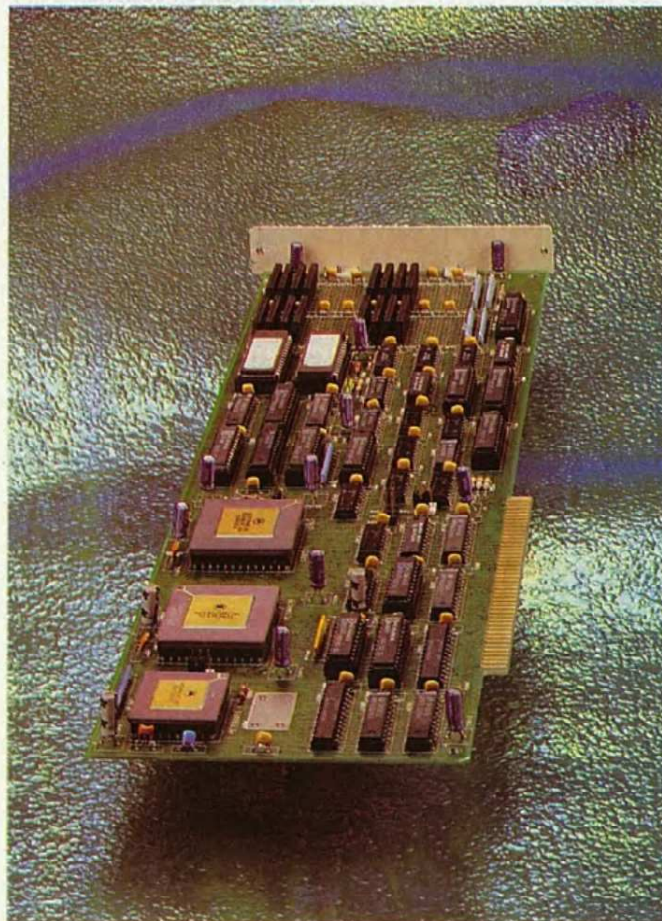
MC68010

Od czasu wprowadzenia procesorów Motoroli serii 68000 rośnie popularność systemu operacyjnego Unix. Ale Unix wymaga możliwości wirtualnego zarządzania pamięcią, a tego MC68000 niestety nie umie. Aby sprostać nowym wymaganiom, opraco-

Podstawy

PROCESORY RODZINY 68000

W każdej Amidze pracuje procesor z rodziny Motorola 680x0. W procesor MC68000 wyposażone są: Amiga 500, 1000 i 2000, zaś w Amidze 3000 zastosowano nowszy i szybszy MC68030. Nie są to jednak wszystkie procesory z tej serii. W kartach przyspieszających, zwanych slangowo "dopalczami" lub kartami turbo, można spotkać także i inne typy procesorów. Postaramy się przedstawić Wam bliżej "rodzinkę" Motoroli. Zamieszczone tu uwagi mogą okazać się przydatne w momencie, gdy będziecie chcieli wymienić procesor zainstalowany w swoim komputerze lub zastosować do niego kartę przyspieszającą.



wany został kolejny procesor Motoroli MC68010 oraz współdziałający z nim układ MMU (Memory Management Unit — jednostka zarządzająca pamięcią), który nazwano MC68451. Jeśli nie weźmiemy na razie pod uwagę MMU, wówczas procesor MC68010 jest trochę ulepszonym procesorem MC68000, w którym dzięki optymalizacji konstrukcji wewnętrznej uzyskano nieco większą szybkość. Pod względem złącz MC68010 jest w pełni zgodny ze swoim poprzednikiem. W zestawie rozkazów występuje jednak pewna różnica: użycie przy MC68010 rozkazu MOVE.SR prowadzi do naruszenia przywileju (privilege violation). Wynika to z tego, że rozkaz ten w nowej Motoroli jest uprzywilejowany i może być wykonany tylko przez program, pracujący w trybie nadzorcy. Wszystkie programy (może poza tymi, które używają wspomnianego rozkazu) działające na Amigach z procesorem MC68000 będą działać również i wtedy, gdy wyposażymy komputer w MC68010. Programy, z którymi będą jakieś kłopoty, można dopasować za pomocą rozwiązań programowych.

Wbrew pozorom wymiana samego procesora nie daje jednak efektu oszałamiającego przyspieszenia, co zobaczycie czarno na białym w tabeli porównawczej "dopalczy". Dlaczego tak się dzieje — będziecie wiedzieli po przeczytaniu dalszego fragmentu tekstu.

MC68020

Jako odpowiedź na wzrastające wymagania ze strony użytkowników już w 1984 roku Motorola wypuściła na rynek MC68020 — jako pierwszy "prawdziwy" 32-bitowy procesor. Zapytacie teraz zapewne, dlaczego zatem w Amidze znalazł się procesor MC68000? Po pierwsze, to, że pierwsza Amiga ukazała się na rynku, nie oznacza jeszcze wcale, że została skonstruowana w ciągu tygodnia. (Dla porównania pierwszy "osobisty" IBM — PC Junior był opracowywany



przez znacznie większy zespół ludzi, mających na dodatek w odróżnieniu od Jaya Minera superkomfortowe warunki — w niesamowitym pośpiechu — "tylko" sześć miesięcy. A efekt?... Szkoda gadać...). Zresztą... nawet jeśli by się dało błyskawicznie dopasować Amigę do MC68020, to koszt tego ostatniego, jako 32-bitowej "nowości", przekraczałby około 20-krotnie cenę Amigi (a pierwsza cena "sklepowej" Amigi 1000 równała się prawie obecnej cenie A3000). Wróćmy jednak do samego proce-

sora. 32-bitowa architektura MC68020 jest już dopracowana we wszystkich szczegółach. Nie tylko rejestry są już w pełni 32-bitowe, ale także szyna danych i szyna adresowa mają 32 bity, a ponadto są już dostępne wszystkie właściwe dla 32 bitów tryby adresowania. Pamięć cache jest zintegrowana z procesorem i nie trzeba jej dokupywać (jak w przypadku procesorów "starszych"). Cache ma 256 bajtów i pozwala na zapamiętanie 128 rozkazów każdego rodzaju. MC68020 może współpracować z koprocetora-
mi matematycznymi MC68881 lub

MC68882, a także może być rozbudowany o MMU MC68851.

MC68030

Kolejnym krokiem naprzód jest procesor MC68030. Wyposażony on jest w dwie niezależnie od siebie pamięci typu cache (każda z nich ma po 256 bajtów). Jedną z nich jest przeznaczona tylko do rozkazów (instruction cache), druga zaś wyłącznie do danych (data cache). MMU — układ zarządzający pamięcią jest wbudowany wewnątrz

MC68030. Zintegrowany układ MMU jest w pełni zgodny z MC68851, jednak nie zawiera wszystkich jego rejestrów i nie ma wszystkich jego funkcji. Szyna pracuje w oparciu o szybki tryb synchroniczny. Nowością jest tu zastosowanie nowego, bardzo szybkiego trybu dostępu do pamięci RAM — Burst.

Koprocесory

W niewielki mikroprocesor nie da się jednak wbudować wszy-
stkiego, co jest niezbędne do

SŁOWNICZEK : KARTY PRZYSPIESZAJĄCE I AMIGA:

Działanie Synchroniczne/Asynchroniczne: Pomieszczone są tutaj trzy, jakie mogą zdarzyć się w świecie reklam Amigi, pojęcia różnych technik hardware'owych. — Omówmy na początku definicję przedmiotu, jaką podaje Motorola. Procesory z serii 68000 mogą współpracować z różnymi rodzajami urządzeń peryferyjnych lub pamięci. W systemach opartych na procesorach tej serii dopuszczalne są cztery tryby pracy. Spójrzmy na koncepcję szyny na przykładzie procesora MC68000:

1. Działanie w pełni asynchroniczne: Omawiane urządzenie współpracuje z procesorem niezależnie od częstotliwości i faz taktowania. Ustawia ono sygnał /DTACK, gdy dane są przygotowywane zarówno do czytania, jak też, gdy są tłumaczone przez proces zapisywania.

2. Działanie pseudo-asynchroniczne: Urządzenie pracuje z taką samą częstotliwością taktowania, jak taktowanie systemu, jednak z własną fazą. Pomiędzy rozpoznaniem sygnału /DTACK procesora a związanym z nim transferem danych występuje przeważnie jeden cykl procesora głównego (CPU). W związku z tym, trzeba ustawić sygnał /DTACK, zanim dane zostaną przełożone na format odczytu lub format zapisu, tak aby ta czynność odbyła się w czasie, jaki upłynie do końca cyklu.

3. Działanie synchroniczne: Wystąpienie sygnału /DTACK i innych asynchronicznych sygnałów wejściowych odbywa się dokładnie w fazie taktowania systemowego.

4. Działanie w trybie Motorola 68000: W tym trybie zachowanie się procesorów rodziny Motorola 68000 zsynchronizowane jest z interfejsem procesora. A zatem, nie bierzemy pod uwagę urządzeń peryferyjnych współpracujących z procesorami tej serii. W tym trybie MC68000 ma zaimplementowane trzy dodatkowe sygnały (E, /VPA i /VMA). Synchroniczny cykl procesora Motorola 68000 nie jest ustawiany wcześniej niż trzeci z opisanych sygnałów.

— W przypadku Amigi 500/1000/2000 taktowanie dodatkowych koprocесorów użytkownika i procesora jest dzielone za pomocą oscylatora kwarcowego 28,37516 MHz. Także i w Amidze 3000 koprocесory te pracują z taką samą częstotliwością. Podwyższenie taktowania (uzyskane na przykład przez wymianę oscylatora) w nadziei zwiększenia szybkości obliczeń jest niemożliwe, ponieważ przy zwiększeniu szybkości taktowania tych koprocесorów psuje się "timing" wizji, a także przestaje funkcjonować kontroler stacji dysków. Jedynie szybszy procesor lub karta przyspieszająca z wyższym taktowaniem mogą skutecznie to rozwiązać.

Przy zastosowaniu kart przyspieszających pozostałe urządzenia dodatkowe do Amigi dają dwie możliwości. Reklamy producenta mówią przede wszystkim o dwóch rozwiązaniach:

1. Rozwiązanie synchroniczne: Tutaj taktowanie procesora na karcie przyspieszającej jest dokładnie sfazowane z taktowaniem systemowym Amigi i jest równe wielokrotności częstotliwości podstawowej (np. podwojona — czyli 14,18 MHz). Wszystkie ważne sygnały sterujące procesora są przez logikę dopasowującą ze strony Amigi tak obsłużone, że dla urządzeń zewnętrznych pozostaje w dalszym ciągu procesor MC68000 pracujący z częstotliwością 7,09 MHz. Zaletą tego rozwiązania jest łatwe podłączenie ze strony Amigi (dzięki dokładnemu sfazowaniu). Wadą zaś jest to, że częstotliwości mogą wynosić jedynie 7,09; 14,18 i 28,37 MHz.

2. Rozwiązanie asynchroniczne: Takt procesora na karcie przyspieszającej jest w pełni niezależny od taktowania systemowego Amigi. Aby dopasować kartę do Amigi pracującej z częstotliwością 7,09 MHz, niezbędne jest dopasowanie hardware'owe. Zaletą takiego rozwiązania: maksymalne taktowanie procesora zależy jedynie od samej karty (np. 50 MHz).

— Należy dokładnie przeczytać wszelkie materiały na temat karty i upewnić się, czy informacja o pracy synchronicznej lub asynchronicznej nie odnosi się przypadkiem do procesora i koprocесora matematycznego na karcie turbo. Koprocесor ten może także pracować synchronicznie (z takim samym taktowaniem jak procesor) lub asynchronicznie (z dodatkowym oscylatorem).

Dynamiczna struktura szyny: 32-bitowe procesory MC68020 i MC68030 różnią się od starszych MC68000 i MC68010 między innymi szerokością szyny i elastycznym interfejsem tej szyny. Przyłożenie sygnału /DTACK jest traktowane przez procesory 32-bitowe jako /DSACK0 i /DSACK1. Za pomocą tych sygnałów można określić od strony hardware, czy transfer danych ma być 8-, 16- lub 32-bitowy, a także, czy cykle oczekiwania (Waitstate) będą potrzebne. A zatem dopasowanie może być do każdego cyklu inne. W związku z tym urządzenia zewnętrzne w takim przypadku "wiedzą", ile jest lub ile może być przeniesionych bajtów, ponieważ ilość ta podana jest przez SIZE0 i SIZE1. W ten sposób procesory 32-bitowe mogą dopasować się do stosowanego sprzętu. Jest to najlepszy sposób na bezproblemowe dopasowanie się do systemu opartego na MC68000. Procesor MC68040 nie ma takiej dynamicznej struktury.

Tryb Burst: cykl Burst potrafi przemieścić podczas dostępu do szyny nie tylko pojedyncze wartości z pamięci, lecz więcej danych, na dodatek z wyjątkowo dużą szybkością. Cykle Burst ustawiają szynę danych o szerokości nawet 32 bitów i mogą być zastosowane w procesorach MC68030 i MC68040. Sygnał handshake /CBREQ /CBACK, a także /STERM "rozumieją" ten tryb od strony sprzętowej. W rejestrze sterującym cache (CACR) bit określający tryb Burst musi być ustawiony. Tryb ten można ustawić zarówno dla instrukcji cache, jak i dla danych cache. Podział cyklu Burst podczas pojedynczego dostępu zwany jest Fake-Burst.

Tryb Burst nie jest stosowany jedynie ze względu na szybkie przenoszenie danych, ale także ze względu na współpracę z nowymi rodzajami pamięci dynamicznej (Nibble-RAM, Page-RAM i Static-Columns-RAM). Maksymalnie 4 długie słowa (długie słowo ma długość 32 bitów) można przemieścić bez cyklu oczekiwania podczas pięciu cykli taktowania procesora. Oba pierwsze cykle są tu normalnie zsynchronizowanym cyklem szyny, podczas gdy pozostałe trzy cykle szyny potrzebują tylko jednego cyklu taktowania zegara. Mówi się, że jest to cykl: 2-1-1-1-Burst. Przy normalnym transferze danych potrzeba na tę samą operację 8 (w trybie synchronicznym) lub 12 (w trybie asynchronicznym) cykli taktowania. W przypadku MC68030 z taktowaniem 20 MHz odpowiada to transferowi przy trybie 2-1-1-1-Burst rzędu 64 MB/s. Gdy pamięć karty turbo nie jest wystarczająco szybka, można zastosować inne szybkości trybu Burst, jak 4-1-1-1 lub 3-2-2-2 przez opóźnienie sygnału /STERM.

Aby było to wyraźnie zrozumiałe, należy zrelatywizować określenie "pełnego Burst". Według danych Motorola określone jest to w sposób następujący:

— gdy pamięć 32-bitowa umożliwi jeden dostęp podczas dwóch cykli taktowania, wówczas może być zainstalowany tryb Burst. Aktywne będą zarówno cache, jak i układy odpowiedzialne za zapis.

— podczas trybu Burst nie może występować żaden cykl oczekiwania (na przykład: 3-2-2-2 lub 4-2-2-2), ponieważ zmniejsza to prędkość przepływu danych (o ok. 10 procent).

różnych jego za-stosowań. Ponadto, nie ma sensu rozbudowywać procesora o funkcje, które są potrzebne tylko niektórym użytkownikom i to do specyficznych celów. Aby przyspieszyć obliczenia (szczególnie na liczbach zmiennoprzecinkowych) — wymyślono koprocesor matematyczny. Oczywiście można to zrobić drogą programową, ale o wiele wygodniej i pewniej jest jednak to zrobić za pomocą dodatkowego sprzętu. Koprocesor pozwala ponadto na działanie na dowolnym typie danych, co rozwiązanie programowe nie zawsze może zapewnić. Dla serii Motorola 68000 zasadniczo istnieją dwa rodzaje koprocesorów:

MC68881

Wszystkie mikroprocesory z serii Motorola 68000 (aż do MC68030) przystosowane są do arytmetyki stałoprzecinkowej. W przypadku obliczeń na liczbach zmiennoprzecinkowych — muszą być one

najpierw "przełożone" na zrozumiałe dla procesora format, a wynik obliczeń też należy "przetłumaczyć". Nie przyspiesza to, rzecz jasna, obliczeń. Można wprawdzie za pomocą odpowiedniej emulacji programowej przeprowadzać obliczenia na liczbach rzeczywistych (zmiennoprzecinkowych), spowoduje to jednak, nie zawsze pożądane — znaczne zwolnienie szybkości obliczeń. Na dodatek — przydałby się układ potrafiący zoptymalizować takie obliczenia. MC68881 jest właśnie takim układem. Czasami koprocesor ten zwany jest także FPU (Floating Point Unit — jednostka zmiennoprzecinkowa). Wszystkie procesory z rodziny Motorola 68000 porozumiewają się z nim przez interfejs. Istnieje jednak pewien problem. System operacyjny Amigi nie potrafi bezpośrednio wykorzystać koprocesora, bowiem wię-

szość popularnych programów (Workbench czy programy użytkowe) nie wykorzystuje FPU i nie ma odpowiednich do tego procedur. Tylko przy wykorzystaniu wersji oprogramowania specjalnie opracowanych na koprocesor można zauważyć różnicę w szybkości obliczeń. Ponadto — koncepcja połączenia procesora i koprocesora matematycznego nie jest w Amidze zbyt dobrze dopracowana. Dostawca koprocesora powinien zadbać o dołączenie do niego odpowiednich bibliotek, których brak w zestawie fabrycznym, sterujących przepływem danych pomiędzy procesorem głównym a koprocesorem matematycznym. Jeszcze słabiej będzie widoczne przyspieszenie obliczeń w przypadku zastosowania kombinacji procesora MC68000 i koprocesora MC68881 w dużej części programów użytkowych. Istnieją wprawdzie wersje tych programów prze-

znaczone do współpracy z koprocesorem (na przykład Imagine, Reflections czy VistaPro), jednak "nastawione" są one na procesor MC68020 i próba zastosowania opisanej wyżej kombinacji procesorów w niektórych przypadkach powoduje nawet zwolnienie (!) obliczeń. Przy "normalnych" wersjach programów koprocesor daje jedynie niezauważalny wzrost szybkości obliczeń (zwalnia go głównie transfer danych).

Siłą koprocesora widać wówczas, gdy użyjemy go razem z procesorami MC68020 czy MC68030. Podłączenie jest proste, wymaga jednak dodatkowej (niewielkiej i niedrożej) przeróbki w zakresie hardware. Do użytkownika należy dozorowanie kodów funkcji i innych połączeń adresowych oraz aktywizacja MC68881 za pomocą odpowiedniego rozkazu, zaś resztę zrobi komputer.

Przy decyzji o wyposażeniu swojej Amigi w koprocesor

SŁOWNICZEK : PAMIĘĆ

Autokonfiguracja: Firma Commodore zastosowała w Amidze specyficzną koncepcję szyny. Podczas włączenia lub resetu wszystkie rozszerzenia i urządzenia zewnętrzne są automatycznie rozpoznawane poprzez wczytywanie danych konfiguracyjnych. Ponadto, obszar adresowy można podzielić na części, które będą później odpowiedzialne za pracę poszczególnych urządzeń. Procesy następujące podczas autokonfiguracji są dość skomplikowane. Są one sterowane za pomocą CONFIGIN i CONFIGOUT. Nie skonfigurowana wcześniej karta zareaguje jedynie wówczas, gdy przyłożymy sygnał niskiego poziomu CONFIGIN, a jednocześnie uzyskany będzie dostęp w obszarze adresowym od \$E80000 do \$E8FFFF. Sygnały pobierane są w A500/A1000 z Expansion Portu, zaś w pozostałych Amigach ze złącz Zorro-II (lub ze slotu MMU). Podczas autokonfiguracji adresy i pozostałe informacje (jak na przykład wielkość pamięci czy dane producenta) zostają uporządkowane i przesłane z Amigi do struktur systemowych. Jeżeli program użytkowy potrzebuje takich informacji, może je w każdej chwili stamtąd pobrać. Duże problemy z autokonfiguracją powstają jednak przy zastosowaniu rozszerzeń pamięci, które są wkładane w podstawkę procesora głównego. Według wytycznych technicznych firmy Commodore, w takim przypadku nie powinien być przenoszony żaden sygnał autokonfigurujący, ponieważ takiego sygnału na złączach wtyczki procesora nie ma. (W jaki sposób zatem działa takie rozszerzenie — pozostaje słodką tajemnicą producenta — przyp. red.) Sprawa ta dotyczy także pamięci RAM na kartach przyspieszających montowanych w ten sam sposób. Producenci takich urządzeń prawdopodobnie "wymuszają" na systemie inny sposób niż przewidzieli to twórcy Amigi. W niektórych kartach odbywa się to za pomocą odpowiedniego ustawienia pamięci w startup-sequence (RAM zostanie w ten sposób rozpoznany i ustawiony w sposób "nieautomatyczny"). Możliwe jest także "oszukanie" Amigi przy tych rozwiązaniach w sposób hardware'owy. Przykładowo hardware karty turbo czeka na adres w zakresie \$E80000 do \$E8FFFF i jednocześnie "łapie" sygnał /AS z procesora. Autokonfigurujące się rozszerzenia w opisanych powyżej "prawidłowych" gniazdach nie reagują w taki sposób, bowiem sam adres bez aktywnego sygnału /AS nie jest wiele wart. "Oszukująca" karta melduje jednak większą pamięć i teraz można już przejść do normalnego trybu pracy Amigi. Rozwiązanie takie również nie jest akceptowane oficjalnie przez Commodore, a ponadto może spowodować problemy w pracy innych urządzeń zewnętrznych przyłączonych do Amigi. Oczywiście producenci tych ostatnich urządzeń zawsze w takim przypadku "zwalają winę" na rozszerzeniu pamięci włożone do podstawki procesora.

Cykle oczekiwania: Procesory Motorola serii 68000 dopasowują się w sposób dość elastyczny do szybkości współpracującej z nimi pamięci roboczej. Procesor używa sygnałów /DTACK (MC68000, MC68010) lub /DSACK0 i /DSACK1 (MC68020, MC68030), podczas gdy RAM zapisuje lub odczytuje dane. Wolne karty RAM odpowiedni sygnał aktywizuje później niż uczyniłby to w przypadku szybkich kart. Odstęp czasowy pomiędzy aktywizacją (sygnał /AS), a sygnałem /DTACK czy /DSACK0/DSACK1 jest czasem oczekiwania dla procesora. Ilość cykli taktowania procesora podczas czasu jego oczekiwania nazywa się cyklem oczekiwania (Waitstate). Jeśli producent określa, że jego urządzenie ma 0 takich cykli — oznacza to, że sygnały /DTACK lub odpowiednio /DSACK zostaną ustawione w tym samym cyklu procesora, co sygnał /AS. Jest to najlepszy z możliwych przypadków (jednak nie należy zapominać o tym, że im mniej cykli oczekiwania ma na przykład karta rozszerzenia pamięci, tym jest droższa). Jasnym jest, że na przykład rozszerzenia pamięci 16-bitowej stawiają pod tym względem o wiele mniejsze wymagania niż włożone do tych samych gniazd rozszerzenia pamięci 32-bitowej. Oznacza to, że jeśli 16-bitowe rozszerzenie pamięci włożone do złącza Zorro-II Amigi wyposażonej w mikroprocesor MC68000 z taktowaniem 7 MHz wymaga 0 cykli oczekiwania, to przy jednoczesnej pracy przyspieszenia o taktowaniu 14 MHz warunek ten może nie być spełniony. Jeśli szybki procesor pracuje z pamięcią o takiej samej częstotliwości taktowania, wówczas musi średnio oczekiwać przez 1 cykl. Jeszcze gorzej przedstawia się sprawa w kartce z "prawdziwą kartą przyspieszającą" wyposażoną w pełny 32-bitowy procesor. Ale to nie wszystko. Przykładowo MC68020 wymaga w asynchronicznym trybie pracy tylko 3 cykli (bez Waitstate) dla przeczytania słowa 16-bitowego, natomiast MC68000 aż czterech. Dla Amigi z taktowaniem 7,09 MHz daje to cykl szyny o długości 564 ns, zaś karta przyspieszająca z procesorem o taktowaniu 14,18 MHz potrzebuje na to samo tylko 212 ns.



matematyczny należy wziąć pod uwagę także maksymalną częstotliwość taktowania, z jaką koprocesor ten może pracować. Procesor główny i koprocesor mogą pracować z tą samą częstotliwością taktowania (tryb synchroniczny) lub z różnymi częstotliwościami (tryb asynchroniczny). Przy tym ostatnim trybie potrzebny jest jednak dodatkowy oscylator kwarcowy do koprocesora. Powinien mieć on możliwość wyboru jednej z dwóch częstotliwości taktowania zegara, dostępnych w takim przypadku.

MC68882

Koprocesor 68882 jest nowszą wersją koprocesora opisanego powyżej — w pełni zgodną ze swoim poprzednikiem. Pracuje z dokładnością 80 bitów. Inne formaty cyfrowe muszą być wcześniej przekonwertowane. MC68882 oprócz części "liczącej" zawiera w sobie również pełny konwerter logiczny, który pozwala na niezależną od obliczeń konwersję dopasowującą format cyfrowy do wymagane-go. Pozwala to na około 1,5 raza większą szybkość działania niż w przypadku MC68881.

MC68851

Wspominany już wielokrotnie wcześniej układ zarządzający pamięcią (MMU) MC68851 jest koprocesorem zwiększającym efektywność działania procesora MC68020 poprzez:

- konwersję adresów logicznych (wykorzystywanych w programach) na adresy "fizyczne" w dostępnej pamięci (tzw. pamięć wirtualna),

- sprawdzanie, czy możliwy jest bezpośredni dostęp do pamięci. (MMU może zabezpieczyć fragment pamięci przed zapisem lub odczytem),

- sprawdzenie, czy bit "nadzorcy" (supervisor) umożliwia dostęp do tej pamięci,

- możliwość definiowania w cache wszystkich adresów, które mają nazwę. Dane przychodzące z portu wejściowego nie mogą być tłumaczone przez cache. Jednak, przy zastosowaniu

MMU, podczas następnego wprowadzenia będą one pobierane nie z portu wejścia, lecz z cache.

Procesor 68851 pracuje "stronami", stąd czasem jest nazywany również PMMU (Paged Memory Management Unit — jednostka zarządzająca stronicowaną pamięcią). Adresowanie odbywa się "stronami". Jedna "strona" odpowiada wielkością bloku pamięci. Użytkownik może określić wielkość takiego bloku z zakresu od 256 bajtów do 32 KB (tylko wartości, które są potęgami liczby 2). W systemie operacyjnym Amigi wszystkie strony pamięci mają tę samą wielkość.

Mikroprocesory Motorola z serii 68000 pracują oczywiście także bez MMU. Bezwarunkowo będzie jednak on wymagany przy wykorzystywaniu pamięci wirtualnej, tam gdzie logiczny obszar adresowania jest większy niż "normalna", fizycznie dostępna pamięć. Umożliwia to uruchamianie największych i najbardziej pamięćochłonnych programów bez kłopotu i kosztów zakupu kości RAM. Pamięć wirtualną można podzielić na dwie podstawowe części — fizycznie dostępną w komputerze pamięć i pamięć dodatkową, której zawartość jest przechowywana na twardym dysku. Wykonywany program znajduje się w pamięci komputera. W momencie, gdy powstanie konieczność dostępu do adresu, który nie leży w tej pamięci, program nadzorczy poinformuje o tym system operacyjny i koprocesor PMMU. Dane zawarte w pamięci komputera zostaną zapisane na twardym dysku, a na ich miejsce zostanie załadowany inny fragment pamięci z poszukiwaną częścią programu. Przy okazji tablice konwersji PMMU zostaną odpowiednio zmodyfikowane i wykonywany program będzie "miał wrażenie", że znajduje się pod zupełnie innymi adresami, niż jest w rzeczywistości.

Tak na przykład jest instalowany KickStart w A3000. Przy uruchomieniu komputera jest on kopiowany do RAM-u, a następnie po modyfikacji tablic konwersji MMU

system pracuje tak, jakby się znajdował w przestrzeni adresowej ROM. (Naprawdę znajduje się w kilkakrotnie szybszej 32-bitowej pamięci typu fast.) W przypadku karty turbo do na przykład A2000 czy A500 istnieje możliwość skopiowania 16-bitowego ROM-u z Kickstartem do 32-bitowej pamięci karty turbo, tak jak na A3000. Ta operacja przyspiesza pracę systemu kilkakrotnie. Szybkość pracy można także zmieniać, włączając i wyłączając cache. W tym celu można skorzystać z programu o nazwie "cpu" z Workbench 2.0 lub programu Public Domain "setcpu" z dysków Freda Fisha. W przypadku komputera A3000 włączenie obu pamięci cache (dla rozkazów procesora i dla danych) zwiększa szybkość z około 8 prędkości A500 do 9-10.

Niestety — firma Commodore nie zadbała dotąd o włączenie do systemu operacyjnego Amigi programu zarządzającego pamięcią wirtualną. Jeśli chcemy ją wykorzystywać, wówczas musimy kupić program z innego źródła. Możliwość wykorzystywania pamięci wirtualnej mają między innymi użytkownicy twardego dysku Evolution firmy Macrossystem. Także firma BSC oferuje program obsługi pamięci wirtualnej zwany "Giga Mem" (około 150 DM) — zarządzający przestrzenią adresową 1 GB, pracujący właściwie z każdym typem twardego dysku.

MC68EC30

Jeśli układ PMMU nie jest nam bezwarunkowo potrzebny do naszych celów (co ma miejsce na przykład w przypadku wykorzystywania Amigi do operacji nie wymagających dużej pamięci), wówczas możemy z oferty Motoroli wybrać uproszczony procesor 68030, czyli MC68EC30 (EC oznacza tu Embedded Controller — "częściowy" kontroler, a niektórzy tłumaczą jako Economic — wersja ekonomiczna). Jest on o wiele tańszy niż "pełny" MC68030 i w pełni zgodny z nim pod względem możliwości zastosowania dodatkowego sprzętu, lecz

zamiast PMMU ma wbudowany uproszczony układ tłumaczący adresy. W konsekwencji, nie można przepisać zawartości ROM-u Kickstartu do szybkiej 32-bitowej pamięci karty przyspieszającej oraz nie można korzystać z pamięci wirtualnej. Dla przeciętnego użytkownika Amigi nie ma to dużego znaczenia, bowiem "zwykłe" oprogramowanie na ten komputer działa z MC68EC30 prawie tak samo szybko, jak z pełnym MC68030.

MC68040

U podstaw podjęcia prac nad procesorem Motorola MC68040 leżały różne założenia integracji konstrukcyjnej. W efekcie uzyskano znaczną poprawę funkcjonalności. Między innymi osiągnięto możliwość jednoczesnej pracy kilku procesorów. Pozwala to na jednoczesne wykonywanie nawet 10—15 zadań bez zauważalnego zwolnienia pracy komputera. Dzieje się tak dlatego, że poszczególne rozkazy można podzielić na kilka "podrozkazów". Każdy z tych "podrozkazów" ma własną jednostkę, w której jest opracowywana. Część "całkowitoliczbowa" jest kierowana do sześciostopniowego przetwarzania potokowego, gdzie jest obrabiana z podwójnym taktowaniem. Każdy stopień takiego przetwarzania wykonuje fragment operacji, właściwej dla danego rozkazu. Na przykład: pierwszy stopień czyta rozkaz, drugi dekoduje go, trzeci oblicza adres efektywny itd., aż do stopnia szóstego. Podstawą tej koncepcji jest jednakże to, że gdy jakiś stopień jeszcze pracuje nad danym rozkazem, poprzedni (lub poprzednie) może już zająć się rozkazem następnym. W ten sposób procesor może pracować jednocześnie nawet z sześcioma rozkazami. Na nieznaczące zwolnienie pracy wpływa to, że przy rozkazach skoku dwa pierwsze stopnie przetwarzania muszą być podwójne.

W procesorze MC68040 koprocesor matematyczny jest

wbudowany w procesor główny. Jest on jednak w pełni samodzielny i może pracować jednocześnie nawet z trzema rozkazami zmiennoprzecinkowymi (gdyż posiada trójstopniowe przetwarzanie). Niestety koprocesor wbudowany w MC68040 nie jest w pełni zgodny z "osobnymi" koprocesorami MC68881 i MC68882 i ma zaimplementowane tylko najczęściej używane rozkazy zmiennoprzecinkowe. Pozostałe rozkazy muszą być emulowane za pomocą oprogramowania (najczęściej w postaci bibliotek na dysku).

Do wszystkich kart z procesorem MC68040 na Amigę, jakie testowaliśmy — była dołączona dyskietka zawierająca odpowiednie biblioteki. Były to: albo licencjonowana przez Commodore "68040.library", albo biblioteka specyficzna dla danego producenta.

Podsumowując, MC68040 może pracować jednocześnie nawet z dziewięcioma rozkazami w postaci zmiennoprzecinkowej. W przypadku rozkazów odwotujących się do pamięci, dane muszą najpierw zostać pobrane z pamięci i po dokonaniu operacji odesłane z powrotem. Aby uzyskać wystarczającą szybkość pracy, oddzielono więc w MC68040 obróbkę rozkazów i danych. W związku z tym potrzebne są dwie jednostki zarządzające (MMU). Przy takim rozwiązaniu nie da się uzyskać 100% zgodności ani z "zewnętrznym" PMU68851, ani z PMMU wbudowanym w procesor MC68030. Producenci kart z procesorem MC68040 dostarczają zatem dyskietkę ze specjalnym oprogramowaniem umożliwiającym uzyskanie takiej zgodności. Rozdzielenie danych i rozkazów procesora wymusiło także zaprojektowanie dwóch oddzielnych pamięci typu cache. Zwiększono także ich pojemność do 4 KB. W odróżnieniu od stosowanej w MC68030 metody dostępu do cache danych (WRITETHROUGH) w MC68040 mamy możliwość dynamicznego zaprogramowania dla każdej "strony pamięci", czyli bloku adresowego:

- odłączenia cache;
- specjalny (sekwencyjny) dostęp do urządzeń wejścia/wyjścia (wyłączony cache dla danych) — dane muszą być czytane lub zapisywane z zachowaniem ściśle określonego porządku. Jest to bardzo ważne w przypadku operacji I/O na przykład dla portu serial;

- metoda Write Through. Metoda ta polega na tym, że każda modyfikacja zawartości danych cache powoduje automatycznie zapisanie zawartości cache do głównej pamięci;

- copy back. Jest to nowy tryb pracy dla cache. Polega on na tym, że cache w przypadku modyfikacji jej zawartości nie jest, jak w przypadku trybu Write Through natychmiast zapisywana do głównej pamięci. Zapis ten jest dokonywany tylko w razie konieczności.

Ważną cechą MC68040 jest także zastosowanie przetwarzania zapisu na szynę danych i to, że można programowo wpływać na to przetwarzanie. W tym przypadku jednostka zajmująca się liczbami stałoprzecinkowymi przekazuje słowo wprost do tego przetwarzania. W momencie, gdy szyna zewnętrzna zwolni się, wówczas zapisane tak dane przekazywane są na szynę, a stamtąd do pamięci.

* Karty przyspieszające z procesorem MC68040 są w tej chwili produkowane (prawdopodobnie ze względu na ich cenę) głównie do A3000 i A2000, choć zdarzają się i do "pięćsetki". Jest to jednak chyba tylko namiastka. Na piorunujący efekt przyjdzie nam jednak poczekać jeszcze trochę do momentu, gdy Commodore rzuci na rynek zapowiadaną od dawna Amigę 4000 z procesorem MC68040 na pokładzie. Jeśli ktoś jest szerzej zainteresowany przedstawionym tematem — polecamy literaturę uzupełniającą w języku niemieckim i angielskim. (Od redakcji: Po polsku na ten temat ukazała się do tej pory jedynie jedna pozycja: skrypt Jacka Kostrzewskiego "Motorola 68000, budowa i lista rozkazów mikroprocesora".

AMIGA Magazin 8/92
Tłum. Marek Pampuch

Czyżby szybciej?

PRZETESTUJMY

Karty przyspieszające (zwane też "kartami turbo" lub slangowo "dopalaczami") dają w przypadku Amigi bardzo dużo. Jednak, nawet jeśli zdecydujecie się na ich zakup — staniecie przed kłopotem związanym z bogactwem wyboru. Poznajmy zatem najpierw dokładnie ich możliwości, wykonanie, ceny i sposoby ewentualnego rozszerzenia. Jak jednak wrzucić je wszystkie do jednego testowego worka?

Michael Eckert,
Stephan Quinkertz

Zamieszaliśmy tym razem nieco w Magazynie AMIGA. Podjęliśmy heroiczny bój zarówno ze "zwykłymi" przyspieszaczami procesora MC68000, jak i z kartami przyspieszającymi wyposażonymi w procesor MC68040. Ten pierwszy rodzaj "dopalaczy" potrafi przyspieszyć Amigę maksimum dwukrotnie, podczas gdy ten drugi przenosi ją do klasy Workstation. Czy jednak najszybsza karta będzie najlepsza? A może nie szybkość jest tu najważniejsza?

Spójrzmy na całą sprawę z nieco innego punktu widzenia. Jak na przykład ma się Volkswagen Polo do Porsche, Mercedesa czy Rolls-Royce'a? Volkswagony nie jest wcale taki zły, ale daje mniejszy komfort jazdy niż Mercedes, jest wolniejszy od Porsche i nie nobilituje tak swojego właściciela, jak Rolls-Royce. Z drugiej strony na nic innego niż Polo Cię nie stać, a przy zakupie decydują nie tylko chęci, ale głównie zawartość portfela.

Podobnie jak przy wyborze samochodu, tak i w przypadku, gdy zdecydujecie się na "dopalacz" — macie dużo możliwości. Przy zakupie należy zatem ocenić zakres, w jakim zamierzamy używać takiej karty, zgodność programową (vel kompatybilność), funkcjonalność, możliwości dal-

szego rozszerzenia (zarówno karty, jak i samej Amigi, w której taka karta zostanie zainstalowana), jakość wykonania samej karty, jej dokumentację, no i oczywiście cenę.

Staraliśmy się przetestować dokładnie wszystkie karty, zarówno pod systemem operacyjnym 1.3, jak i 2.0, stawiając parametry pracy rozkazem "cpu cache burst fastrom" (lub innym o podobnym działaniu. Rozkaz ten włącza pamięć cache procesora, tryb Burst obsługi RAM i kopiuje zawartość ROM-u Amigi do 32-bitowej pamięci RAM.) Programy porównujące (Benchmark AIBB i testy praktyczne) były uruchamiane z poziomu Workbench 2.0 (Kickstart w wersji 37.175, Workbench w wersji 37.67). W tabelkach, które są podsumowaniem testów, znajdziecie nie "bezpośrednie" wyniki testów, lecz porównanie szybkości osiąganych przez Amigę pracującą z kartami przyspieszającymi i nie wyposażone w te karty. Poziomym odniesienia były:

- dla kart do A500:
 - Amiga 500 (wersja 6A, 1 MB Chip RAM-u, rozszerzenie pamięci Golem (wyposażone w 2 MB pamięci), a przy kartach, które wymagały większej pamięci — Golem RAM Box z 4 MB);
 - dla kart do A2000:
 - Amiga 2000 (wersja 6.2, 1 MB Chip RAM, 16-bitowe rozszerzenie pamięci Golem RAM 2000 wyposażone w 2



MB pamięci (0 cykli oczekiwania – waitstate);
oraz

— Amiga 3000 (wersja 6.1, model z procesorem o zegarze 25 MHz, z 2 MB Chip RAM i 8 MB Fast RAM);

Szybkość procesora to nie wszystko

A oto najważniejsze kryteria, jakie przyjęliśmy przy testach:

* Osiągi — aby określić możliwości kart przyspieszających, użyliśmy nie jednego programu, lecz zestawu programów:

— Test AIBB — opisany dalej program porównawczy Public Domain o nazwie AIBB zawiera różne testy, które określają osiągi systemu zarówno dla procedur całkowitoliczbowych, jak i dla zmiennoprzecinkowych. Podano średnią wyników, jednak zaznaczono także poszczególne osiągi kart, celem wyraźniejszego rozgraniczenia ich słabych i silnych stron. Zaletą takiego rozwiązania jest to, że możecie porównać w domu wybrane osiągnięcia Waszej Amigi z wynikami naszych testów.

Przeprowadzanie pojedynczych testów za pomocą programu AIBB i podawanie później średnich wyników ma jednak także pewną wadę. Jeśli na przykład uruchomimy pojedynczy test (na przykład "Write Pixel"), możemy uzyskać doskonały wynik, może jednak się zdarzyć, że dla innego testu przy tej samej karcie uzyskamy wynik bardzo słaby, jednak średnia wyjdzie i tak nieco wyższa niż dla innej karty, która w obu przypadkach da wynik średni. To, którą z tych dwóch kart ocenimy wyżej, zależy od tego, do czego głównie stosujecie swoją Amigę.

* Test praktyczny — tutaj do oceny osiągnięć karty użyliśmy 5 różnych popularnych programów z różnych zakresów działania komputera:

— Imagine 2.0 — ten program animacyjny bazujący na ray-tracingu był używany w dwóch wersjach. Wersja całkowitoliczbowo stosowana

była do testowania zwykłych przyspieszaczy procesora MC68000 oraz kart turbo, które nie były wyposażone w koprocesor matematyczny, natomiast wersja zmiennoprzecinkowa programu wchodziła w grę wówczas, gdy na kartach był zainstalowany ten koprocesor. W obu przypadkach do testu użyliśmy zbioru o danych do obiektu o nazwie Camaro.new (jest to samochód na fragmencie szachownicy, 320 x 512 punktów). Ten obiekt został przez nas przeliczony w trybie trace do grafiki 24-bitowej (24-bitowy ILBM). Obliczono średnią uzyskanych wartości dla obu wersji programu tak, że są reprezentatywne dla obu wersji programu Imagine.

— AD Pro 2.0 — Art Department Professional (w skrócie AD Pro) jest programem do obróbki obrazu. Podczas testu opracowywaliśmy grafikę 24-bitową (obrazek "Czereśnie") traktując ją funkcją Convolve. Przy stosowaniu operatora Convolve włączany jest przez program płaski filtr liniowy, przy czym charakterystykę filtra można umieścić w macierzy 3 x 3 lub 5 x 5. Do testu wykorzystaliśmy macierz "wyostrzenia obrazu" (Sharpen) 5 x 5, która jest cyfrowym filtrem wygładzającym niektóre nieostrości rysunku.

Z testów z wykorzystaniem tych dwóch programów wynika pierwszy wniosek: jeśli macie zamiar pracować dużo z grafiką i wideo oraz z innymi programami, wymagającymi wielu obliczeń — wybrana przez Was karta turbo powinna charakteryzować się jak najwyższym współczynnikiem szybkości. Pozostałe parametry w tym przypadku odgrywają nieco mniejszą rolę.

— Aquarium 1.15 — to program Public Domain (z dysku Fisha nr 559). Jest to specjalizowana baza danych, zawierająca informacje o dyskach PD Freda Fisha. Używaliśmy Aquarium z danymi o prawie całym zestawie z tej serii (od dysku nr 1 do dysku nr 600). Wyszukiwaliśmy na nich łańcuch tekstowy "Charlie Heath" występujący zarówno na pierwszym, jak i na kilku dalszych dyskach. Mierzenie czasu za-

kończyliśmy po sprawdzeniu zawartości dysku 600. Program i odpowiedni zestaw danych został wcześniej przekopiowany do RAM-dysku. Szybkość dostępu do pamięci miała jednak wpływ na wynik, podobnie jak i w kilku następnych testach.

— Beckertext II — najczęściej używany w Niemczech program do redagowania tekstów również może skorzystać na tym, że wyposażysz swoją Amigę w "dopalacz". W teście zmierzaliśmy czas przesuwania na ekranie (scroll) tekstu ASCII (bez znaków niemieckich) o objętości 35 KB (co daje około 24 stron ekranowych). Następnie poleciliśmy programowi, by wszystkie napotkane w tym tekście litery "e" zastępowane były przez "ee". Czas obu tych operacji został zsumowany. Wniosek: Jeśli chcecie pracować z programami do obróbki tekstów i edytorami — wówczas powinniście wybierać takie karty, które umożliwiają najszybszy przepływ danych.

— Lattice C — jak zapewne wielu z Was wie, kompilacja kodu źródłowego może nieraz zabrać sporo czasu. Dla potrzeb testu skopiowaliśmy wszystkie potrzebne dane (kod źródłowy programu w języku C, kompilator, linker, dane do "inkludów" itd.) do RAM-dysku, a następnie kompilowaliśmy kod źródłowy i linkowaliśmy go za pomocą kompilatora Lattice C (wersja 5.1 b). Mierzony był czas całkowitej kompilacji i linkowania programu. Podobnie jak przy teście bazy danych — i tutaj czas dostępu do RAM-u miał wpływ na uzyskany wynik.

— Workbench 2.0 — System operacyjny w tej wersji (używaliśmy WB 2.04) daje większy komfort pracy i ma lepsze osiągi niż jego poprzednik. Postanowiliśmy zbadać, jak się ma sprawa z otwieraniem "okien". Workbench 2.0 w tym zakresie pracuje nieco wolniej niż wersja 1.3. Karta turbo powinna jednak ten proces znacząco przyspieszyć. Spróbowaliśmy to zbadać. W tym celu przekopiowaliśmy do RAM-dysku katalog z 200 identycznymi podkatalogami. W każdym z tych podkatalogów

znajdowało się po 10 procedur NoFastMem wraz z ikonami. Następnie potraktowaliśmy całość opcją Select Contents (wybierz zawartość) z menu "Window" (znajdującego się na listwie tytułowej Workbench). Z kolei otworzyliśmy (opcją Open z menu "Icons" wszystkie 200 podkatalogów po kolei, tak aby na ekranie pokazała się ich zawartość, czyli ikony) i zmierzaliśmy całkowity czas tej operacji. Ten test wykorzystywał prawie całą zastosowaną w naszych testowanych komputerach pamięć Chip. W związku z tym, w opisanym przypadku dużą rolę grał także czas dostępu do pamięci Fast (gdzieś przecież musieliśmy założyć RAM-dysk). W rezultacie uzyskaliśmy wyraźne przyspieszenie, i to zarówno przy najprostszych "przyspieszacach", jak i przy najbardziej wyrafinowanych kartach.

— DiskSpeed 4.1 — karty turbo potrafią także znacznie przyspieszyć szybkość operacji dyskowych Amigi (i to zarówno przy stacjach dyskietek, jak i przy twardych dyskach). Jeśli mówimy o kontrolerach (choć przy kontrolerach SCSI bardziej na miejscu byłoby określenie "Host Adapter"), musimy wziąć pod uwagę dwa podstawowe tryby działania:

— DMA: kontroler przenosi dane z dysku do pamięci Amigi bez korzystania z procesora głównego,

— PIO (lub CPU) — cały proces przenoszenia danych z dysku do pamięci jest sterowany przez procesor.

Aby określić osiągnięcia przyspieszaczy w tym zakresie, używaliśmy kontrolera pracującego w trybie PIO (jak łatwo się domyślić, w tym przypadku wpływ "dopalacza" będzie znacząco większy. Był to kontroler Golem SCSI-2 z dyskiem twardym Quantum LPS 105 S). Wybór trybu pracy kontrolera jest zależny od tego, do czego chcemy używać twardego dysku, i w tym przypadku powyższe zdanie nie jest żadną kryptoreklamą ani zaleceniem przy zakupie kontrolera. Po prostu taki mieliśmy pod ręką. W przypadku testów dla A3000 wykorzystywany był kontroler

zamontowany fabrycznie i taki sam dysk 105 MB. Jeśli karta przyspieszająca miała własny kontroler, wówczas używaliśmy go. Szybkość przepływu danych została zmierzona za pomocą programu PD o nazwie DiskSpeed (wersja 4.1), z dysku Fisha nr 574. W tabelkach z wynikiem testu znajduje się wskaźnik obliczony na podstawie odczytu "Long Aligned". Testy zostały przeprowadzone na pustej partycji o wielkości 50 MB i obejmowały odczyt danych testowych o wielkości 4096 bajtów.

* Instalacja: 32-bitowa pamięć RAM karty turbo może być skonfigurowana zarówno jako pamięć rozszerzona (Extended) lub jako autokonfigurowana. System automatycznie rozpoznaje, jak ma skonfigurować pamięć, ale na wszelki wypadek — powinna istnieć możliwość skonfigurowania pamięci przez użytkownika.

* Kompatybilność: Większa część produkowanych obecnie urządzeń dodatkowych nie sprawia kłopotów przy współpracy z kartami turbo, jednak nie zawsze tak jest. W związku z powyższym, przetestowaliśmy współdziałanie kart przyspieszających z różnymi urządzeniami (jak na przykład: rozszerzenia pamięci, dyski pracujące w standardzie DMA i PIO, genlocki itp.). Wybraliśmy do

testów te urządzenia, które w testach przeprowadzanych wcześniej przez AMIGA Magazyn uzyskały oceny bardzo dobre. Sprawdzając zgodność w zakresie programów — użyliśmy gier (które ze względu na niechlujne programowanie zazwyczaj sprawiają najwięcej problemów użytkownikom kart turbo). Ponadto przetestowaliśmy karty z różnymi wersjami Amig (A2000 — wersje 4.2 i 6.2, A500/500+ — wersje 5A, 6A i 8.1, A3000 — wersje 6.1 i 7).

* Dokumentacja: Uznaliśmy, że dokumentacja w języku niemieckim jest w każdym przypadku czynnikiem pozytywnym (jeśli idzie o rynek polski, to chyba jest dokładnie odwrotnie, a zatem w tym wypadku wynik testów może być, z "polskiego" punktu widzenia nieco zniekształcony — przyp. red.). Przy ocenie dokumentacji najpierw postawiliśmy sobie pytanie: "Jakie cechy powinna mieć dobra dokumentacja?" Naszym zdaniem dobra dokumentacja:

— nie powinna pozostawiać korzystającemu z niej żadnych wątpliwości. Sposób instalacji, ustawienia przełączników (jumperów) oraz sposób obsługi programów instalacyjnych i testujących kartę po instalacji powinny być w niej dokładnie opisane; — powinna "pasować" do

karty. Może to wydać się śmieszne, ale spotkaliśmy się z przypadkami, w których instrukcja dotyczyła innego modelu karty. A to już nie jest takie śmieszne — zwłaszcza dla początkującego użytkownika;

— powinna zawierać opis innych możliwości karty (jak na przykład możliwość jej ewentualnej rozbudowy o koprocessor, inny procesor główny itp.). Pożądany jest spis treści i krótkie podsumowanie instrukcji;

— powinna zawierać rysunki lub zdjęcia pomagające w instalacji karty czy przy ustawieniu jumperów;

— zrozumienie sposobu działania karty znacznie ułatwia przykładowe programy wraz z dokładnymi objaśnieniami.

Oprócz wymienionych czynników dużą rolę odgrywa jakość instrukcji. Kilka naprędce skserowanych kartek, połączonych jednym spinaczem, na pewno nie jest tym, o czym marzy użytkownik, a ponadto nadaje się praktycznie tylko do jednorazowego przeczytania.

* Jakość wykonania karty: Dokładnie wytrawiona płytka o "przemysłowej" jakości i ze starannie wykonanymi punktami lutowniczymi, przelotkami, a także z dokładnym i "cynoooszczędnym" lutowaniem ze zrozumiałych względów

uzyska w teście wyższe oceny niż płytka wykonana mniej starannie. Nie jest to jedynie kwestia odczuć estetycznych. Połączone styki i kontakty zapewnią prawidłową pracę karty przez długi czas. Niektóre tańsze rozwiązania nie mają tych cech. W takim przypadku utlenianie się styków może spowodować problemy z kontakowaniem. Duża zaletą są prawidłowo dobrane (nie za luźne) podstawki pod układy scalone.

* Ocena końcowa: Na całkowitą ocenę danej karty składają się:

— wygoda obsługi i instalacji (x 1);

— jakość wykonania (x 1);

— dokumentacja (x 2);

— osiągi (x 3) — przez osiągi rozumiemy oceniane niezależnie od kosztu karty wyniki wszystkich testów (Benchmark AIBB, testy praktyczne, kompatybilność, sposób instalacji, jakość wykonania i dokumentacji oraz możliwość długotrwałej pracy);

— wartość karty (to znaczy stosunek ceny do osiągnięć) — jest to ocena wszystkich wyżej opisanych czynników z uwzględnieniem ceny karty przyspieszającej (nie zawsze bowiem opłaca się "przepłacić" za lepiej wykonaną kartę).

AMIGA Magazyn 8/92
Tłum. Marek Pampuch

Ogłoszenia drobne

Ogłoszenia drobne ukazują się na specjalnych stronach w standardowej postaci — po dwanaście na stronie. Cena jednego ogłoszenia wynosi 350 tys. złotych. Chcąc opublikować tego typu ogłoszenie, należy dokonać wpłaty na konto Wydawnictwa (IX Oddział PKO BP w Warszawie r-k 1599-318121-136, nazwa rachunku: LUPUS Sp. z o.o. Warszawa, ul. Stępińska 22/30), a następnie dołączyć kopię przekazu do kartki z ogłoszeniem składającym się z nagłówka (np. nazwa firmy albo nazwa produktu) o długości nie przekraczającej 15 znaków oraz treści nie przekraczającej 450 znaków — i całość wysłać pocztą do redakcji. Rzecz jasna załatwić całą sprawę i zapłacić można także na miejscu, w redakcji. Uwaga: nie będą przyjmowane ogłoszenia drobne w formie do reprodukcji — ani w całości, ani w części (np. znaki firmowe).

PUBLIC DOMAIN!

Oferujemy serie Fish i inne.

Nagrane dyskietki 3.5 NN po 22 tys., od 10 szt. po 19 tys., od 50 szt. po 16 tys., od 400 szt. po 14 tys.

Nagranie na dyskach 3.5" Verbatim Datalife - droższe o 3 tys., a na 5.25" NN tańsze o 4 tys.

Proszę zamawiać robiąc przekaz na adres:

Eureka ul. Żwirki i Wigury 13a
62-300 WRZESŃNIA,

a w miejscu na

korespondencję podawać treść zamówienia.

Polecamy dyski Fish od numeru 500. Proszę przysłać 15 tys. po katalog.

DYMAREX

UL. MESSNERA 14/1

03-982 WARSAWA

tel. 15-93-38

wykonuje i wysyła do AMIGI :

Sampler Mono 229

Sampler Stereo 329

MIDI 249

przełącznik mysz-joystick 69

inne akcesoria :

przejście, aby w NINTENDO grać typowym, dużym joystickiem

Nr 1 (sterujące) 89

Nr 2 79

kable do AMIGI, ATARI i ST interfejsy do ZX Spectrum testery joysticków (również IBM)

ceny w tys. złotych ważne w roku 1992

Dieter Meyer

Karty przyspieszają się ze zrozumiałych powodów "przechwalone". Każdy producent za pomocą różnych sztuczek stara się zamaskować ich ewentualne słabe strony. Jeśli chcemy dokonać obiektywnej oceny wartości kart — powinniśmy dać wszystkim testowanym urządzeniom jednakowe szanse. Taką "wspólną platformą" są programy nazywane Benchmark. Za ich pomocą możesz szybko sprawdzić obiektywną wartość cech danego urządzenia. Do opisywanych w tym numerze testów kart turbo wybraliśmy program Public Domain — AIBB (Amiga Intuition Based Benchmark). Jak sama nazwa wskazuje, jest to "zorientowany na graficzny interfejs użytkownika (w przypadku Amigi — Intuition) program testujący".

Po uruchomieniu programu z poziomu Workbench lub CLI/Shell — mamy przez chwilę wrażenie, że coś z naszą Amigą jest nie w porządku. Ekran robi się czarny, i pojawia się na nim napis "Please wait — evaluation system" (proszę czekać — ocena systemu). Nie jest to jednak znany wszystkim Guru Message — zatem nerwowe naciskanie klawiszy myszki nic w tym przypadku nie pomoże. Należy chwilę poczekać. Gdy program ustali już potrzebne mu parametry systemu i częstotliwość taktowania — pojawi się wygodny, obsługiwany za pomocą myszki ekran główny. Omówmy jego zawartość, zaczynając tradycyjnie od lewego górnego rogu.

Performance Graph — jest to graficzne przedstawienie w postaci "wykresu słupkowego" wartości zmierzonych w aktualnie przeprowadzanym teście. Dla porównania masz tu kolejno osiągnięcia, jakie uzyskałoby w tym teście kilka różnych komputerów (domyślnie są to od lewej: Twoja Amiga, A500 z 512 KB (bez pamięci typu Fast, A2000 z 1 MB pamięci typu Fast, A2500 z procesorem o taktowaniu 20 MHz i

Program testujący Benchmark AIBB

STOPER

Określenie osiągnięć systemu komputerowego nie sprawia ostatnio żadnych trudności. Służą do tego programy należące do grupy określanej ogólną nazwą Benchmark. Dosyć wygodnym rozwiązaniem jest przedstawiony poniżej program Public Domain o nazwie AIBB.

A3000 z procesorem o taktowaniu 25 MHz). Można je zapamiętać jako porównawcze wartości osiągnięte w testach przez inne modele Amig. Najlepszy osiągnięty wynik jest paskiem w kolorze czerwonym, pozostałe paski są żółte.

Test Result/Information — to okno zawiera różne dane. Najpierw widzicie tu nazwę testu, następnie aktualnie uzyskany wynik. Poniżej pojawiają się liczbowe wyniki testu.

Base Machine Indication — tutaj wypisana jest nazwa i konfiguracja (koprocessor — FPU, procesor zarządzający pamięcią — MMU) Amigi, która jest traktowana w teście jako punkt odniesienia.

Comparison Information — tu podane są dane, które określają cyfrowo wskaźnik porównania szybkości dla komputerów przedstawionych na wykresie słupkowym, a także nazwę przeprowadzanego testu.

Basic Information — poniżej wykresu słupkowego i określenia Amigi traktowanej jako poziom odniesienia znajdują się dane opisujące testowany komputer.

Test Activation Gadgets — w prawym dolnym rogu znajduje się szereg gadżetów. Przez wybranie za pomocą myszki dowolnego z nich spowodujemy uruchomienie testu o nazwie wypisanej na gadżecie. Test zostanie przeprowadzony z wcześniej ustawionymi parametrami. Procedury testujące są podzielone na dwie grupy "standardowe" (opisy jasnoniebieskie) oraz "zmiennoprzecinkowe" (opisy żółte).

■ Przez wybranie opcji z odpowiedniego menu, rozwijanego na liście tytułowej, można dokładnie określić parametry potrzebne do danego testu.

MENU 1: General

ABOUT AIBB podaje informacje o autorach i numerze wersji tego programu.

ENTER/EXIT HELP MODE — jeśli tryb Help jest włączony — wybranie tej opcji powoduje ukazanie się na ekranie okna z informacjami pomocniczymi. Ponowny wybór tej opcji spowoduje powrót do ekranu głównego.

SAVE CONFIGURATION — zapisuje aktualnie ustawione parametry jako poziom porównawczy dla testów, jakie będą przeprowadzone po kolejnym uruchomieniu programu AIBB.

QUIT — powoduje zakończenie programu.

MENU 2: Systems

SYSTEM INFORMATION — otwiera menu pomocnicze, z danymi różnych konfiguracji systemowych, które są zapisane na dysku i mogą zostać z niego wczytane. Najciekawsze z tego wszystkiego wydają się dane zgrupowane pod nazwą "This Machine" dające Wam szczegółowe określenie konfiguracji Waszej Amigi.

AIBB TASK PRIORITY — tutaj podany jest priorytet, z którym pracuje program AIBB w stosunku do innych programów, jakie są aktualnie wykonywane w komputerze. Kolejne opcje menu można wybrać tylko wówczas, gdy procesor testowanego kom-

putera ma możliwość pracy w danym trybie.

SWITCH INSTRUCTION CACHE — włącza lub wyłącza pamięć typu cache przeznaczoną na rozkazy. Aktualny stan widoczny jest na ekranie głównym (Inst Cache).

SWITCH DATA CACHE — podobna funkcja, jak opisana powyżej, odnosząca się jednakże do cache dla danych.

SWITCH I-CACHE BURST MODE — włącza lub wyłącza tryb Burst dla pamięci cache przeznaczonej na rozkazy.

SWITCH D-CACHE BURST MODE — jak wyżej, tyle że dla danych cache.

SWITCH 040 COPYBACK MODE — włącza lub wyłącza tryb Copy Back dla procesora MC68040.

MENU 3: Test Options

DISABLE MULTITASKING — wyłącza całkowicie wielozadaniowość, aby zmniejszyć wpływ równoległe wykonywanych programów na dokładność testu.

SCREEN OVERLAY — przełącza ekran monochromatyczny (2 kolory, 1 bitplan, niska rozdzielczość) na "normalny" ekran, na jakim pracuje program AIBB (16 kolorów, 4 bitplany, wysoka rozdzielczość). Jest to opcja, która ma zastosowanie tylko podczas testów.

SET COMPARISON BASE — za pomocą tej opcji możemy wybrać dowolną z zapisanych na dysku konfiguracji systemowych i ustawić jako punkt odniesienia dla aktualnie testowanej Amigi.

STANDARD 68000 CODE — ustawia testy tak, aby były kompatybilne z wszystkimi procesorami serii 68000.

68020+ CODE — dopasowuje testy do rozszerzonych możliwości procesorów 68020, 68030 i 68040. Opcja ta nie jest dostępna w przypadku Amig wyposażonych w procesory MC68000 lub 68010.

STANDARD MATH CODE — emuluje programowo procedury obliczeniowe koprocesora matematycznego. W ten sposób można dokonać porównania karty przyspieszającej bez koprocesora i z koprocesorem.

IN-LINE COPROCESSOR CODE — jeśli wybierzesz tę opcję, wówczas możesz testować specjalne możliwości, jakie dają Ci koprocesory matematyczne. Te dwie ostatnie możliwości wpływają tylko na te testy, które bazują na obliczeniach zmiennooprzecinkowych. Wszystkie pozostałe opcje menu są ważne dla wszystkich testów.

MENU 4: Special

ENTER/EXIT REVIEW MODE — za pomocą tej opcji można wszystkie testy zafundować do pamięci. Jeśli ten tryb jest uaktywniony i wybierzesz określone wyniki jednego z poprzednich testów, wówczas przez wywołanie danego testu zostaną wyświetlone uzyskane poprzednio wyniki, dotyczące zarówno wywołanego testu, jak i wszystkich następnych. Ponowny wybór tej opcji spowoduje powrót do ekranu głównego.

START/STOP LOG FILE — po wybraniu tej opcji program zaczyna prowadzić "dziennik pomiarowy" dotyczący wszystkich przeprowadzanych testów. Ich nazwy i wyniki zostaną zapisane w zbiorze na dysku.

ALL TESTS/MAKE MODULE — ta opcja pozwala na utworzenie modułu obejmującego tylko wybrane testy.

Przejdźmy teraz do opisanie poszczególnych testów. Najpierw te, które pracują bez wykorzystania koprocesora matematycznego:

WRITE PIXEL — otwiera nowe okno w niskiej rozdziel-

czości i zapełnia je określonym kolorem. Podczas testów używane są procedury systemowe SetAPen() i WritePixel().

DHRYSTONE — test spotykany w prawie wszystkich programach typu Benchmark. Algorytm oblicza szybkość działania procesora i podaje ją w dhrystonach na sekundę.

MATRIX — symuluje różne obliczenia na trzech macierzach o wymiarach 50 x 50. Ten test wymaga dodatkowo 29,3 KB pamięci.

MEMTEST — test działający na pamięci, która zostanie podzielona na bloki pamięci typu Fast i Chip. Następnie ustalone bloki zostaną przepisane z Fast-RAM-u do Chip-RAM-u, z Chip-RAM-u do Chip-RAM-u i z Fast-RAM-u do Fast-RAM-u. Test wymaga 32 KB Chip-RAM-u i 32 KB Fast-RAM-u.

SIEVE — jest to procedura znana pod nazwą "Sito Erastotenesa". Oblicza liczby pierwsze w ustawionym wcześniej zakresie.

SORT — sortuje zestaw 30000 16-bitowych liczb całkowitych. Test ten potrzebuje dodatkowo 58,6 KB pamięci.

IMATH — używa wielu funkcji matematycznych (całkowitoliczbowych), między innymi czterech podstawowych działań (mnożenie, dzielenie, dodawanie i odejmowanie).

TGTEST — wykorzystuje procedury graficzne systemu operacyjnego. Wprowadzony wcześniej tekst zostaje wyświetlony na ekranie, a następnie przesunięty.

Następne testy dotyczą kart turbo, które wyposażone są w koprocesor matematyczny. Przy użyciu procesora MC68040 należy wziąć pod uwagę to, że koprocesor wbudowany jest w procesor główny. Taki wbudowany koprocesor może obsługiwać tylko podstawowe funkcje matematyczne. Funkcje transcendentne (na przykład sinus, cosinus i tangens) nie są obsługiwane, a zatem takie funkcje przy testach muszą być emulowane programowo. Program pomocniczy tego typu jest dołączony do wszystkich kart przyspieszających wyposażonych w procesor 68040. Aby funkcje transcendentne mogły być

użyte do testów, obowiązuje poniższa konwencja:

NONE: gdy funkcje transcendentne nie są używane,

LIGHT: jeśli 5-20% obliczeń wykorzystuje te funkcje,

MODERATE: gdy 21-50% obliczeń wymaga funkcji transcendentnych,

HEAVY: gdy funkcje transcendentne stanowią więcej niż 50% obliczeń,

FMATH: bardzo podobny test do opisanego powyżej IMATH, lecz wykorzystujący operacje zmiennooprzecinkowe zamiast całkowitoliczbowych (tu ustawiamy parametr NONE według opisanej przed chwilą konwencji).

SAVAGE: używa szybko przełączalnych funkcji transcendentnych (HEAVY),

FMATRIX: taki sam test jak "standardowy" test macierzowy, z tą różnicą, że użyte tu zostaną operacje zmiennooprzecinkowe (NONE). Ten test wymaga dodatkowo 37,5 KB pamięci.

SWHETSTONE i **DWHETSTONE**: różnią się tylko tym, że SWHETSTONE pracuje z liczbami pojedynczej precyzji, zaś DWHETSTONE — z liczbami o precyzji podwójnej (MODERATE).

BEACHBALL: rysuje na ekranie piłą "plażową" składającą się z cieniowanych kwadratów (w rozdzielczości 640 x 400 punktów, 16 kolorach i w trybie Interlace). Tu ustawiamy parametr LIGHT,

FTRACE: używa procedur używanych zazwyczaj przez programy ray-tracingowe (LIGHT),

CPIXTEST: seria obliczeń liczb zespolonych (LIGHT lub MODERATE),

Program AIBB można używać za pomocą modemu z Compuserve w grupie AMIGAUSER (go:AMIGAUSER) w bibliotece 15 (nazwa zbioru danych: AIBB41.lzh). Czytelnicy niemieckojęzycznego AMIGA Magazin mogą otrzymać program AIBB w wersji 4.2 na podstawie pisemnego zezwolenia autora (LaMonte Kopp) na dyskietce "serwisowej" do numeru 8/92. Wersja 4.2 została ulepszona w stosunku do opisywanej wersji 4.1 w kilku punktach, jednak otrzymywane wyniki są takie same.

AMIGA Magazin 8/92
Tłum. Marek Pampuch

Christian Seiler

Michael Eckert

Karty przyspieszające z procesorami MC68030 i MC68040 są dosyć kosztowną "dokładką" do Amigi. Nie każdy może wyłożyć ponad 10 milionów złotych po to, aby zmusić swoją Amigę do szybszej pracy. Z drugiej strony "zwykle" przyspieszacz procesora MC68000 można nabyć już za kwotę w granicach 1,5 miliona złotych. Sprawdziliśmy, czy te tanie karty są rzeczywiście tak dobre, jak wygląda to w reklamach producentów.

* AVANTI — do tej karty, która jest wkładana do podstawki procesora możecie oprócz taktowanego z szybkością 14.18 MHz procesora MC68000-12 włożyć także koprocesor matematyczny MC68881 lub MC68882. Koprocesor może być taktowany synchronicznie z procesorem lub asynchronicznie, przy wykorzystaniu dodatkowego wykroczenia.

Dokumentacja/Instalacja: Dość kiepska instrukcja w języku niemieckim składa się z obustronnie zadrukowanej kartki formatu A-4. Instalacja tej karty jest kuriozalna: Testowaliśmy kilka Amig (wersje 3, 5, 6A i 8.1), lecz nie znaleźliśmy ani jednej takiej, do której by karta Avanti pasowała. Po prostu, po włożeniu karty w podstawkę procesora klawiatura nie chciała się zamknąć, bowiem zahaczała o procesor. Na szczęście plastikowa obudowa Amigi jest dosyć elastyczna, więc w końcu po wielu próbach jakoś nam się udało. Po zainstalowaniu tej karty nie można jednak zamocować płytki przełączającej systemu operacyjnego. Przy instalacji karty eliminującej Interlace typu "flicker-flicker" (również wkładanej w podstawkę procesora) nie było żadnego z opisywanych wyżej problemów. Ale to jeszcze nie wszystko. Standardowe biblioteki Workbench "nie widzą" koprocesora z karty Avanti. Aby "zobaczyć", należy dokupić dodatkowo zestaw bibliotek o nazwie FPU-Explore-Set (co kosztuje kolejne



700.000 zł) i uaktywnić je przez dopisanie w sekwencji startowej rozkazu "Initialise". Aby dopasować kartę Avanti do posiadanej Amigi, należy odpowiednio przestawić jumper. Za pomocą przełącznika można wybierać pomiędzy 7- a 14- megahercowym taktowaniem procesora.

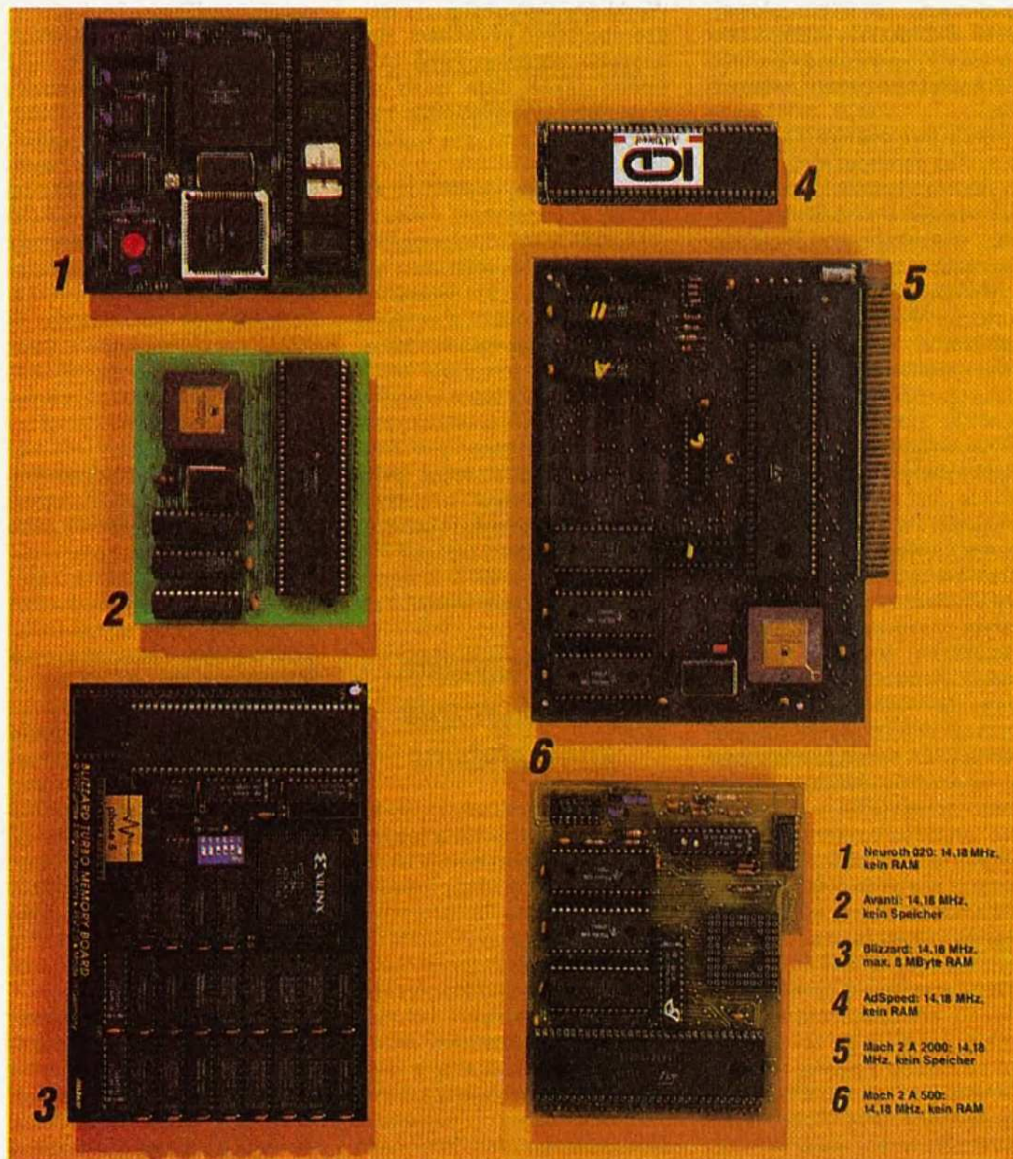
Kompatybilność/jakość wykonania: Podczas pracy z różnymi rozszerzeniami pamięci karta Avanti nie stwarzała większych problemów. Odnosiło się to także do kontrolerów twardego dysku wyposażonych w dodatkową pamięć. Przy niektórych kontrolerach twardego dysku było jednak różnie. Kontrolery "GVP A500+" i "Supra 500XP" działały bez problemów, zaś "Golem-SCSI-2-System" działał wolniej niż na "zwykłej" Amidze (nie wyposażonej w kartę Avanti) — jak wykazały wyniki testów przeprowadzonych różnymi programami typu "benchmark", zwłaszcza w trybie 14 MHz. Rozwiązanie tej zagadki jest następujące: Karta Avanti pracuje wprawdzie z taktowaniem 14 MHz, jednak przy dostępie do hardware (na przykład stacja dyskietek) przełącza się z powrotem na 7 MHz. Nie jest to rozwiązanie najszcześniejsze, ponieważ urządzenia sterowane przez ODD-CIA (czyli na łączu szeregowym) pracują (na "normalnej" Amidze) z nieco wyższym taktowaniem. Ponadto w tym przypadku Event-Timer chodzi nieco szybciej, niż to przyjęto w większości programów testujących. Taktowanie w A500 jest pochodną sygnału synchronizacji pionowej i stąd to całe zamieszanie. Przy Amidze 2000 taktowanie jest synchronizowane z wykorzystaniem zasilacza, a zatem taki błąd nie powinien wystąpić. Aby sprawdzić tę teorię — przetestowaliśmy kartę Avanti z Amigą 2000. Porównanie wyników testu w trybie pracy z taktowaniem 7 MHz wykazało, że nie ma tu takich różnic, jakie występowały w przypadku A500. Wydaje nam się, że producent powinien rozwiązać ten problem za pomocą małego programu dopasowującego.

Wystarczy 1,8 miliona

Jeśli założymy koprocessor, wówczas zaczną funkcjonować nowe biblioteki matematyczne. Przyjęte jest, że oprogramowanie powinno korzystać z bibliotek matematy-

Do Amigi 500 / 2000

KARTY 68000/68020



- 1 Neuroth 020: 14,18 MHz, kein RAM
- 2 Avanti: 14,18 MHz, kein Speicher
- 3 Blizzard: 14,18 MHz, max. 8 MByte RAM
- 4 AdSpeed: 14,18 MHz, kein RAM
- 5 Mach 2 A 2000: 14,18 MHz, kein Speicher
- 6 Mach 2 A 500: 14,18 MHz, kein RAM

cznych z katalogu "libs", co jest zaimplementowane nawet w programach Public Domain typu "mandelbrot". Jednak wersje zmiennoprzecinkowe większości programów komercyjnych (na przykład Imagine czy Reflections) wymagają równoczesnego zainstalowania szybszego procesora (co najmniej MC68020). Jeśli wystartujemy z taką wersją programu bez wymiany procesora MC6800-12, to pomimo tego, że koprocessor matematyczny jest zainsta-

lowany — zostaniemy natychmiast uraczeni ekranem z Guru Meditation. Wersje całkowitoliczbowe tych programów zadziałają, ale przyrost szybkości będzie ograniczony. Karta Avanti ma wszystkie układy scalone na podstawkach. Pod procesorem znajdują się dwa prowizorycznie połączone układy.

Stosunek osiągnięty do ceny: Opisane problemy, wykryte podczas niezbyt długiego testu pozwalają spodziewać się jeszcze gorszych rzeczy pod-

czas eksploatacji. Mniej niż przeciętne osiągi karty Avanti są warte jedynie omówienia. Nie radzimy natomiast jej kupować

Avanti (14,18 MHz, bez koprocessora), cena około 1,8 mln złotych. W wersji z koprocessorem i zestawem programów FPU-Explorer-Set — około 4,5 mln złotych.

* ADSPEED — producent AdSpeed nie oparł się tylko na procesorze MC68000-12 taktowanym z prędkością 14,18 MHz, lecz dodał także 32 KB

pamięci SRAM umieszczonej na maledkiej płytce. Pamięć ta gra rolę pseudo-cache i również trochę się o to, aby komputer pracował szybciej. Można zmieniać szybkość pomiędzy 7 a 14 MHz zarówno na drodze programowej, jak i sprzętowej. Jeśli wybierzemy niższe taktowanie, wówczas pamięć SRAM jest po prostu odłączana.

Dokumentacja/Instalacja: Ta ostatnia jest prosta. Wystarczy tylko włożyć płytke w podstawkę procesora i gotowe. Instrukcja po angielsku jest skąpa, ale wystarczająca.

Kompatybilność/jakość wykonania: Płytkę AdSpeed jest dokładnie wykonana i niewiele większa od samego procesora, który umieszczony jest w podstawie. Reszta płytki wykonana jest techniką SMD. Dzięki tak zwartej budowie AdSpeed nie zastania miejsca, w którym chcemy zainstalować przełącznik systemów operacyjnych lub kartę "flicker-flicker". Z pozostałym hardware (z jednym wyjątkiem) nie było problemów. Kontroler dysku twardego GVP A500+ używa do przesyłania danych za pomocą DMA swojego własnego rozszerzenia pamięci. Choć takie przesyłanie nie zatrudnia procesora, kontroler nie sygnalizuje wysyłania na zewnątrz żadnego cyklu DMA, w związku z czym cache AdSpeeda "nie zauważa" przesyłania danych. Przy kolejnym dostępie do pamięci AdSpeed odczytuje stare dane znajdujące się w cache, co powoduje zawieszenie się komputera. Ten błąd został zauważony przy kontrolerze GVP z ROM-em w wersji 3.12. Aby uniknąć pomyłki, należy uruchamiać specjalny program z parametrem icdcache. Ponadto AdSpeed musi wystartować z taktowaniem 7.14 MHz, a dopiero później zostać programowo przełączony na taktowanie 14.18 MHz. Przy testach z rozszerzeniami pamięci bez tego kontrolera nie było żadnych problemów. Wbudowanie karty w A500 (wersja 3, 5, 6A) i A500 Plus (wersja 8.1) idzie dosyć ciężko (procesor znów zawadza o klawiaturę), ale w porównaniu z Avanti — AdSpeed pasuje

lepiej.

Osiągi/cena: AdSpeed nie pozwala na osiągnięcie magicznego wskaźnika dwukrotnego przyspieszenia. Przy wykorzystywaniu niektórych funkcji systemowych (Write Pixel, TGTest) czy pracy z Workbenchem następuje dalsze obniżenie szybkości. Relatywnie wysoka cena AdSpeed także daje dużo do myślenia i powstrzymuje nas od podania porad dla kupujących.

AdSpeed (14,18 MHz, 32 KB cache) — cena około 4 mln zł.

* MACH 2 — w tym przypadku połączono możliwości Avanti i AdSpeed. Mach 2 (wcześniejsza nazwa: Tomado) ma 16 KB pamięci SRAM jako pseudo-cache i podstawkę na koprocessor matematyczny. Karta oferowana jest w wersji wkładanej w podstawkę procesora A500, a także jako karta wkładana do slotu Amigi 2000. Podobnie jak Avanti — Mach 2 wymaga dodatkowego oprogramowania. Producent oferuje FPU-Explorer-Set firmy ECS — wersję dopasowaną do karty Mach 2.

Nie wszystko jest faktycznie kompatybilne

Dokumentacja/instalacja: skąpa instrukcja w języku niemieckim (dla wersji wkładanej w podstawkę procesora) składa się z 3 kartek formatu A-4 obrazujących instalację i ustawienie jumperów. Opisu błędów, rzecz jasna, "zapomniano" w niej umieścić. Testowana przez nas, druga wkładana w slot A2000, wersja miała tylko jednostronicową instrukcję z najważniejszymi informacjami dotyczącymi instalacji i konfiguracji karty. Wskazówek na temat tego, jakie dodatkowe oprogramowanie należy zakupić przy zainstalowaniu koprocessora — w żadnej instrukcji nie było. Oprócz łatwej instalacji polegającej na włożeniu karty w odpowiednie miejsce, użytkownik Mach 2 musi dodatkowo ustawić odpowiednio przełączniki DIP-switch (w wersji do A2000) lub dopasować mostek wtykowy do istniejącej konfiguracji systemu (pamięć, typ układu Ag-

nus). Niektóre układy Agnus nie współpracują dobrze z kartą Mach 2 i muszą być zamienione przez układ GAL.

Kompatybilność/jakość wykonania: Przy kontrolerze GVP wystąpiły w przypadku Mach 2 te same problemy, co przy AdSpeed — po aktywacji cache następowało Guru. W przypadku karty Mach 2 można temu zaradzić, ale tylko przez całkowite wyłączenie wszystkich DIP-switchów. W obu testowanych typach karty Mach 2 wszystkie układy były na podstawkach, jednak staranność wykonania płytki pozostawiała wiele do życzenia. Podczas gdy po instalacji w A2000 nie ma kłopotów, posiadacze A500 (wersja 5, 6A, 8.1) spotykają się z identycznymi problemami, jak przy karcie Avanti, czyli z niedostatecznym dopasowaniem. Mach 2 ma taki sam jak Avanti brak kompatybilności z programami w przypadku zastosowania koprocessora matematycznego. Najgorsze ze wszystkiego jest jednak to, że przy wersji wkładanej w podstawkę procesora (do A500) zmienia się napięcie zasilające kwarc koprocessora. Spróbowałismy zatem pracy karty Mach 2 z synchronicznym taktowaniem procesora. Niestety, nawet "Imagine" nie przeszło tego testu. Po wyjęciu koprocessora spróbowaliśmy wersji "całkowitoliczbowej" tego programu. Po nieco dłuższym okresie pracy program zawiesił komputer.

Stosunek ceny do osiągnięć: Mach 2 we wszystkich testach wykazywał relatywnie stałe osiągi, jednak w żadnym punkcie nie pobił AdSpeeda. Oto mała próbka samooceny producenta wzięta z dokumentacji:

"Mach 2, według dokonanych przez nas pomiarów jest mniej więcej około 26 procent wolniejszy niż karta 68030 produkcji Commodore (chodzi o kartę A2630 z taktowaniem 25 MHz)."

Radzimy wam porównać tę notatkę, z pobieżnym choćby spojrzeniem na tabelkę wyników pomiarów naszych testów. Zastosowanie koprocessora daje karcie Mach 2 równie niewiele, co w przypadku Avanti.

Porada przy zakupie: Użytkownicy A2000 powinni instalować wyłącznie wersję wkładaną do slotów (MMU). Zastosowanie koprocessora (w obu typach karty Mach 2) powinno zainteresować tylko tych, którzy chcą uzyskać rozszerzone możliwości programowania, a nie zależy im na przyspieszeniu.

Mach 2 (14,18 MHz, bez koprocessora) — cena około 2,8 mln zł. W wersji z koprocessorem i zestawem programów FPU-Explorer-Set — około 4,5 mln złotych.

Różnice w osiągnięciach

BLIZZARD — producent reklamuje to rozszerzenie jako Turbo Memory Board — oznacza to, że ma na myśli szybkie rozszerzenie pamięci. Wygląda to na chwyt marketingowy, a ponadto sieje w umyśle kupującego dodatkową niepewność: "Co to może być takiego ta rozszerzająca pamięć karta turbo?" Przede wszystkim popatrzmy na ten przyspieszacz procesora MC68000: Procesor jest taktowany z 14,18 MHz, tak jak w dotąd opisywanych kartach. Ponadto na płycie znajduje się miejsce na 8 MB pamięci RAM (z reguły wyposażone w 2 MB pamięci i rozszerzalne po 1 MB). Pamięć ta jest podzielona na 5 banków (1 bank to 4 podstawki). Cztery z nich są zarezerwowane pod amigowską pamięć typu Fast RAM, która może być wypełniona pamięciami DRAM ustawionymi 4 x 256 KBit lub 4 x 1 MBit (może występować też mieszana organizacja pamięci). Ostatni bank przeznaczony jest na Shadow RAM (512 KB z pamięciami DRAM 4 x 256 KBit). Do tej ostatniej pamięci można przełączyć system operacyjny z dyskietki lub bezpośrednio z Kickstart-ROM-u. Cała ta pamięć (w tym i Shadow-RAM) pracuje także z taktowaniem 14 MHz bez cykli oczekiwania (wait-state), co daje pewne zwiększenie szybkości. Po załadowaniu systemu operacyjnego pamięć Shadow-RAM zostaje zabezpieczona przed zapisem.

Dokumentacja/Instalacja: Wzorowo przygotowana instrukcja jest dwujęzyczna (angielska i niemiecka) z wieloma przykładami i objaśnieniami. Po jej przeczytaniu instalacja wydaje się dziecinnie łatwa: należy wyjąć stary procesor z podstawki i w pustą podstawkę włożyć kartę. Skonfigurowanie karty (między innymi wielkość pamięci czy taktowanie początkowe) ułatwia sześć DIP-Switchów. Instalacja kości pamięci daje użytkownikowi pełną dowolność, bowiem podstawki nie są związane z któryś z banków (pamięć konfigurowujemy przełącznikami).

Kompatybilność/jakość wykonania: Karta Blizzard nie potrafi zgodnie "współżyć" z niektórymi rozszerzeniami pamięci dla Amigi 500 i A500 Plus. Odnosi się to, na przykład do pamięci RAM wbudowanych w niektóre kontrolery twardego dysku (Multi Evolution, Supra 500XP), przy których nie działa. Jeżeli jednak na płycie kontrolera nie ma żadnej pamięci RAM lub jest ona odłączona, wówczas nie ma żadnych komplikacji. Testy wykazują, że zwłaszcza w starszych kontrolerach Supra pamięć jest przyłączana od adresu \$00200000. Dopóki inne samokonfigurujące się rozszerzenia będą wypełniać obszar poniżej tego adresu, nie będzie żadnych konfliktów adresowych, ponieważ karty leżące "poniżej" będą w dalszym ciągu w innym obszarze adresowym. Samokonfigurująca się pamięć karty Blizzard przyłącza się jednak w tym samym obszarze adresowym co RAM z karty Supra, a to nie może dobrze działać. Kontrolery Golem SCSI 2 i GVP serii II współpracują z pamięcią karty Blizzard bez problemów. Dotyczy to także wewnętrznych kart rozszerzających pamięć o 512 KB oraz kart Big RAM (2.5 MB rozszerzenia pamięci). Pamięć RAM karty Blizzard pracuje z identycznym (zerowym) priorytetem, co ewentualnie dołączone (wolniejsze) karty rozszerzenia pamięci. Prawie każdy program (na przykład AdPro2) będzie chciał zatem najpierw skorzystać z wolniejszej pamięci, co doprowadzi do znacznego pogorszenia osiągnięć. W przeciwieństwie na przykład do karty "flicker-flicker" czy karty AdSpeed — po instalacji

karty Blizzard nie ma miejsca na dołożenie przełącznika Kickstartów. Jakość wykonania płytki jest pokazowa, płytka jest polakierowana, wszystkie układy scalone znajdują się w sprężynujących podstawkach.

Stosunek ceny do osiągnięć: Blizzard był jedyną kartą z grupy "przyspieszaczy 68000", który zarówno w teście Benchmark, jak i w testach praktycznych powodował stałe, dwukrotne przyspieszenie działania Amigi. Biorąc pod uwagę jego podwójną funkcję jako przyspieszenie komputera i rozszerzenie pamięci (oddzielne rozszerzenie z 2 MB pamięci kosztuje około 2.800.000 zł.) otrzymał od nas ocenę bardzo dobrą.

Porady przy zakupie: Jeśli zakupicie kartę Blizzard nie wyposażoną w pamięć Fast-RAM i Shadow-RAM, wówczas osiągi karty będą nieco obniżone. Pamięć Shadow RAM i minimum 2 MB Fast-RAM jest niezbędne. Przy późniejszej rozbudowie możecie zwiększyć ilość pamięci na karcie Blizzard do 8 MB i nie będą Wam już wtedy potrzebne inne karty rozszerzające. Jeżeli posiadacie kontroler twardego dysku Supra-500XP z pamięcią RAM na płycie kontrolera, wówczas musicie odłączyć układy pamięci znajdujące się na płycie karty Blizzard.

Blizzard (14,18 MHz bez pamięci RAM) — około 3.200.000 zł. Blizzard (jw., wyposażony w 512 KB pamięci Shadow-RAM i 2 MB pamięci Fast-RAM) — około 5.000.000 zł.

* NEUROTH 020 — ta karta z procesorem MC68020-16 mieści się w klasie cenowej "przyspieszaczy" procesora MC68000. Jest zatem jasne, że na czymś musiano zaoszczędzić. Producent zrezygnował tu z pamięci 32-bitowej, a także koprocesora zarządzającego tą pamięcią (PMMU MC68851). Na płycie znajduje się jednak podstawka na koprocesor matematyczny, który jest taktowany asynchronicznie z własnego oscylatora kwarcowego. Karta jest wkładana w podstawkę procesora głównego, przy czym wykorzystuje dalej ten

procesor, a za pomocą przełącznika możecie wybrać procesor, z jakim zamierzacie pracować. Jako oprogramowanie dołączona jest jedynie dyskietka z kilkoma programami typu Public Domain.

MC68000 z koprocesorem niewiele daje

Dokumentacja/Instalacja: Nioemieckojęzyczna instrukcja składa się z trzech kartek A4 bez rysunków. Instalacja jest prosta. Procesor MC68000 przekładamy do podstawki na karcie, a w pustą miejsce na płycie wkładamy kartę.

Kompatybilność/jakość wykonania: podczas naszego testowego "młyn" nie wystąpił brak kompatybilności z hardwarem (wewnętrzne karty rozszerzenia pamięci o 512 KB, zewnętrzne rozszerzenie pamięci Golem RAM Box — wyposażone w pełne 8 MB, kontrolery twardego dysku: Golem SCSI 2, GVP A500+ z 4 MB pamięci RAM, Supra 500XP z 2 MB pamięci RAM i pamięcią ROM w wersji AMAB6). Jeśli chodzi o oprogramowanie — występują te same ograniczenia, co w przypadku kart przyspieszających z procesorem MC68030 (wiele gier i niestandardnie napisane programy nie działają z procesorami 32-bitowymi), ale to nie jest sprawą samej karty. Karta jest dokładnie wykonana, kilka programowanych układów logicznych zrobiono techniką SMD. Procesory MC68000 i MC68020 oraz opcjonalny kwarc do koprocesora można umieścić w specjalnych podstawkach, zaś sam koprocesor matematyczny w podstawce typu PLCC. Niestety po zainstalowaniu karty nie ma już miejsca na przełącznik Kickstartów. Inne rozszerzenia wewnętrzne (Multi Vision 500, Big-RAM 25) dają się w tym przypadku zamontować bez trudności.

Cena w stosunku do osiągnięć: bez koprocesora ta karta z procesorem MC68020 osiąga nieco gorsze wyniki niż "zwykłe" przyspieszacze MC68000. Przede wszystkim

bardzo nieznacznie przyspieszone są funkcje systemu operacyjnego. Jest to cena, jaką płacimy za brak pamięci 32-bitowej. Rozczarowują także wyniki testu praktycznego (Aquarium, Becke rtext, kompilator C, Workbench) przy nie zainstalowanym koprocesorze. Przy programie Imagine (bez koprocesora) jest niewiele lepiej (wskaźnik przyspieszenia równy 1,7). Jeśli jednak zainstalujemy koprocesor, wówczas wyniki w zakresie działania i kompatybilności z oprogramowaniem są o wiele lepsze niż przy kombinacji "procesor zwykły i koprocesor" (MC68000/MC68881) zastosowanej w kartach Mach 2 i Avanti. Wszystkie użyte w testach wersje "zmiennoprzecinkowe" programów (Imagine, Vistapro, Turbo Silver, Reflections i kilka innych) "zaakceptowały" kartę z koprocesorem bez zastrzeżeń.

Porady dla kupujących: Jeśli potrzeba wam niedrogi "dopalacza" do pracy z programami, które korzystają z koprocesora — wówczas Neuroth 020 jest rozwiązaniem właściwym, bowiem zakupione za podobną cenę "przyspieszacze 68000" są od niego gorsze. Do wszystkich innych zastosowań karta z procesorem MC68020 nie jest tak całkiem niezbędna.

Neuroth 020 (14,18 MHz, bez koprocesora) — około 3.700.000 zł. Neuroth 020 (jw., z koprocesorem MC68882-16 i 16 MHz oscylatorem kwarcowym) — około 6.000.000 zł.

Z ostatniej chwili: Producent zapowiedział, że w najbliższej przyszłości na rynku ukaże się wersja karty wyposażona w 32-bitową pamięć RAM, jednak w momencie przeprowadzania testów nie dotarła jeszcze do naszej redakcji.

Neuroth 020 (1MB pamięci RAM 32-bitowej, z koprocesorem MC68882-16 i 16 MHz oscylatorem kwarcowym) — około 6.300.000 zł. Neuroth 020 (4MB pamięci RAM 32-bitowej, z koprocesorem MC68882-16 i 16 MHz oscylatorem kwarcowym) — około 8.000.000 zł.



ZESTAWIENIE KOŃCOWE

PODSUMOWANIE

Stephan Quinkertz

Michael Eckert

Testowaliśmy wszystkie karty przyspieszające, jakimi dysponowała nasza redakcja. Spraw-

daliśmy też ponownie te, które były wcześniej osobno oceniane przez (niemiecko-języczny) AMIGA Magazin (na przykład A2620). Te ostatnie sprawdzaliśmy teraz w nieco innych warunkach, po to, aby przybliżyć je Czytelnikom, zwłaszcza że w między-

Największy i wymagający największego nakładu pracy test wszechczasów — test kart turbo — przyniósł bardzo dużo danych. Jak należy interpretować wyniki? Która karta okazała się najlepsza?

czasie producenci powprowadzili do nich wiele zmian. W poniższych tabelkach znajdziecie zestawione wyniki testów. Noty z ewentualnych testów wcześniejszych tracą ważność. Stopniowanie ocen także różni się od wcześniej stosowanego, ale przy kolej-

nych pojedynczych testach nie będziemy go stosować (wrócimy do tego, które stosujemy "normalnie"). Pojawiło się już wiele nowych kart z szybszymi procesorami i koprocesorami. Nie jest jednak powiedziane, że samo podwojenie taktowania procesora

TESTWERTE: 68000/020/030

	A 2000 68000	A 2000 68010	Avanti	Blizzard	Mach 2	AdSpeed	Neuroth 20	A2620	Golem Turbo	VXL-30 40/25	VXL-30 25/25	A 3000 25 MHz
AIBB-TEST												
Write Pixel	25,68 s	1,01	1,09	1,90	1,78	1,19	1,38	2,63	3,29	1,36	1,28	4,41
Sieve	63,60 s	1,07	1,09	2,02	1,95	1,95	4,86	5,10	5,19	13,09	8,55	9,06
Dhrystone	1426 D/s	1,06	1,06	2,02	1,77	1,76	1,19	2,70	3,79	1,37	1,29	5,25
Sort	58,98 s	1,01	1,09	2,01	1,90	1,90	2,47	3,25	4,25	5,35	4,46	7,09
Matrix	22,10 s	1,01	1,15	2,01	1,77	1,78	3,24	5,02	5,70	4,64	4,13	9,36
IMath	95,64 s	1,12	1,29	2,01	1,88	1,88	6,15	9,49	10,83	11,35	8,82	17,78
MemTest	37,46 s	1,01	1,07	1,87	1,70	1,73	3,13	3,70	3,87	3,29	3,04	6,18
TGTest	11,62 s	1,01	1,03	1,46	1,35	1,12	1,15	1,71	1,87	1,11	1,06	1,96
Savage	702,54 s	1,08	1,24	2,01	1,88	1,88	133,06	116,31	118,47	209,09	208,47	209,09
FMATH	72,04 s	1,02	1,05	2,01	1,76	1,76	13,05	10,12	12,84	19,37	17,88	22,37
FMatrix	31,64 s	1,02	1,08	2,02	1,61	1,62	3,18	3,68	4,54	4,62	4,13	7,46
Beachball	1106,56 s	1,08	1,23	2,00	1,87	1,85	18,00	18,93	26,60	22,68	20,28	40,68
SWhetstone	24620 W/s	1,07	1,21	2,01	1,86	1,86	27,81	27,30	39,43	46,38	48,90	65,51
DWhetstone	27012 W/s	1,08	1,22	2,01	1,86	1,86	20,77	25,01	35,00	31,24	27,35	56,58
FTrace	501,32 s	1,08	1,23	2,01	1,87	1,86	55,15	54,97	67,56	76,65	73,08	114,98
CplxTest	140,54 s	1,10	1,18	2,01	1,81	1,80	7,00	12,09	15,62	9,14	8,73	23,90
PRAXISTEST												
Imagine 2.0	24310 s	1,08	1,20	2,08	---	1,87	3,36	3,52	5,40	4,04	3,80	8,71
AdPro 2.0	445 s	1,03	1,07	2,07	1,93	1,89	2,34	3,16	3,94	2,13	1,96	5,43
Aquarium 1.15	296 s	1,09	1,42	2,04	1,72	1,64	1,38	2,99	3,84	1,56	1,55	4,85
Beckertext II	861 s	1,03	1,06	1,92	1,72	1,63	1,48	2,66	3,22	1,56	1,47	3,84
Lattice 5.0	765 s	1,05	1,06	2,08	1,12	1,66	1,35	1,35	1,69	1,49	1,40	5,28
Workbench 2.0	2866 s	1,09	1,05	2,16	1,82	1,51	1,15	2,71	3,27	1,15	1,12	5,42
DiskSpeed 4.1	372 KByte/s	1,02	1,05	1,38	1,19	1,12	1,02	1,28	1,55	0,94	0,93	1,32

TESTWERTE: 68030

	A 2000 68000	GVP S-II 22 MHz	GVP S-II 33 MHz	Impact 25 MHz	Impact 50 MHz	G-Force 030/25EC	G-Force 030/25	A 530 030/40	G-Force 030/40	G-Force 030/50	Prof. 3000 28 MHz	Golem II 32 MHz
AIBB-TEST												
Write Pixel	25,68 s	4,47	6,00	4,83	6,45	2,00	4,65	4,81	6,42	6,79	4,47	5,97
Sieve	63,60 s	8,05	12,00	9,14	17,97	9,11	9,11	14,42	14,52	18,07	10,23	11,73
Dhrystone	1426 D/s	5,82	7,86	6,13	8,99	5,92	5,92	7,72	8,51	9,58	5,25	8,35
Sort	58,98 s	6,58	9,64	7,43	13,47	7,26	7,26	11,02	11,34	13,78	7,64	9,67
Matrix	22,10 s	8,77	12,85	9,87	17,54	9,78	9,78	14,54	15,14	18,11	10,14	12,85
IMath	95,64 s	16,78	24,40	18,75	34,16	18,61	18,61	28,13	28,98	34,65	19,36	24,27
MemTest	37,46 s	5,41	7,37	5,61	8,83	5,83	5,83	5,71	7,97	8,96	6,26	6,94
TGTest	11,62 s	1,99	2,16	2,05	2,19	1,47	2,05	2,00	2,26	2,18	2,06	2,15
Savage	702,54 s	163,91	276,59	209,09	418,18	209,09	209,09	334,54	334,54	418,18	237,34	268,15
FMATH	72,04 s	20,01	29,77	22,65	45,03	22,65	22,65	35,84	36,02	45,03	25,37	29,05
FMatrix	31,64 s	7,03	10,34	7,68	14,25	7,79	7,79	11,76	12,17	14,65	8,15	9,95
Beachball	1106,56 s	39,44	55,55	44,55	72,04	34,56	43,88	52,95	62,31	72,90	45,54	55,49
SWhetstone	24620 W/s	60,80	90,66	67,70	131,87	67,70	67,70	106,58	108,02	131,87	74,66	87,54
DWhetstone	27012 W/s	54,12	79,10	60,89	110,18	60,10	60,10	91,67	93,49	114,26	63,83	78,43
FTrace	501,32 s	107,58	162,77	117,68	236,47	121,09	121,09	185,67	185,67	241,02	127,89	151,92
CplxTest	140,54 s	24,15	34,28	25,93	42,85	25,55	25,55	36,50	38,82	45,63	24,83	34,28
PRAXISTEST												
Imagine 2.0	24310 s	6,45	12,19	9,08	18,00	9,12	9,44	14,22	14,81	17,51	9,74	12,12
AdPro 2.0	445 s	6,10	8,73	6,54	10,80	6,36	6,54	9,08	9,67	11,13	6,27	8,73
Aquarium 1.15	296 s	5,48	7,59	5,92	8,71	4,42	6,04	7,59	8,46	8,71	5,38	7,40
Beckertext II	861 s	4,44	5,91	4,51	5,96	3,99	4,60	5,86	6,06	6,15	4,31	5,38
Lattice 5.0	765 s	5,93	8,05	6,27	9,56	5,35	6,32	8,41	9,33	9,94	5,63	8,14
Workbench 2.0	2866 s	5,29	6,74	5,77	8,07	2,04	5,83	7,75	7,96	8,05	5,16	6,78
DiskSpeed 4.1	372 KByte/s	1,28	1,63	1,80	1,65	1,29	1,50	1,11	1,39	1,91	1,58	1,65



OCENA KART

	A 2630	Rocket Launcher	G-Force 030/25	G-Force 030/EC25	G-Force 030/40	G-Force 030/50	Golem Turbo BI I	Golem Turbo BI II	Professional 3000
Procesor	MC68030-25	MC68030-50	MC68030-25	MC68EC030-25	MC68EC030-40	MC68030-50	MC68030-33	MC68030-33	MC68030-25
Takt (MHz)	25	50	25	25	40	50	32	32	28,36
Koprocesor	MC68882-25	MC68882-50	MC68882-25	MC68882-25	MC68882-40	MC68882-50	MC6888-33	MC68882-33	MC68882-25
Takt (MHz)	25	50	25	25	40	50	32	32	28,36
Pamięć RAM (MB)	4	brak	5	5	4	4	4	4	4
Dodatkowy hardware	brak	brak	kontroler SCSI	kontroler SCSI	kontroler SCSI	kontroler SCSI	brak	kontroler SCSI	brak
Przybliżona cena w markach	1500	1400	2050	1900	2500	3300	2100	2250	2290

NOTOWANIA

Ocena z uwzględnieniem kosztu	dobra	dobra	bardzo dobra	dobra	dobra	bardzo dobra	dobra	bardzo dobra	dostateczna
Dokumentacja	dostateczna	dostateczna	dobra	dobra	dobra	dobra	dostateczna	dostateczna	niedostateczna
Komfort obsługi	dobra	dobra	dobra	dobra	dobra	dobra	dobra	dobra	dostateczna
Jakość wykonania	bardzo dobra	dobra	bardzo dobra	bardzo dobra	bardzo dobra	bardzo dobra	bardzo dobra	bardzo dobra	bardzo dobra
Osiągi	dobra	dobra	dobra	dobra	dobra	bardzo dobra	bardzo dobra	bardzo dobra	dobra
Punktacja (0-12)	9,6	9,5	10,4	9,5	9,5	10,7	9,9	10,2	8,0
Ocena	dobra	dobra	bardzo dobra	dobra	dobra	bardzo dobra	dobra	bardzo dobra	dobra

ra oznacza też podwojenie osiągnięć karty — najlepszym przykładem jest tu Rocket Launcher dla karty A2630. Ponadto, przykładowo, szybkość "prawdziwej" karty 50-MHz może zostać znacznie zwolniona przez resztę elektroniki (na przykład kości pamięci RAM, które mogą nie być dopasowane do szybszego taktowania procesora (w przypadku, gdy mają tzw. cykle oczekiwania — waitstat). W tabelkach znajdziecie też dane dla kart turbo, które nie są już produkowane, a na rynku pojawiają się w

śladowych ilościach (na przykład na wyprzedających), oraz kart najnowszych, których jeszcze w handlu nie ma. W przypadku takich właśnie kart zrezygnowaliśmy z wystawiania ocen.

Osobną kwestią są "dopalacze" z procesorem MC68EC030. Wbudowana w nie namiastka koprocesora zarządzającego PMMU teoretycznie nie powinna działać prawidłowo z Workbenchem 2.04 (związana z rozkazem CPU przy opcji FASTROM — kopiującej ROM do 32-bitowego RAM-u). Tym większe było

nasze zdziwienie, gdy okazało się, że pewna część kart z procesorem MC68EC030-25 lekceważy sobie teorię i jednak działa, prawdopodobnie dysponując w pełni sprawnym PMMU w procesorze głównym.

Odpowiedź uzyskana z firmy Motorola wyjaśniła ten "cud". Po uruchomieniu produkcji MC68EC030 — firma sprzedała tyle tych procesorów, że nie nadążała z ich produkcją. Dlatego też część procesorów MC68030 była sprzedawana jako MC68EC030. Z punktu widze-

nia funkcjonowania procesora nie stanowiło to żadnej różnicy.

MC68030 nie ma żadnego wpływu na działanie funkcji systemowych, natomiast przy procesorze typu EC wpływ ten jest znaczny. Ponadto, ponieważ nie przeprowadzono nigdy dokładniejszych testów jednostek PMMU, firma nie zaleca ich stosowania w systemach opartych na procesorze MC68030. Opisany tu typ EC procesora był produkowany przez krótki czas (do końca roku 1991), dopóki nie sprzedano zadowolającej firmę

TESTWERTE: 68030/040

	A 2000 68000	Stormbr. 50 MHz	Stormbr. 55 MHz	Mega Mid. 38/50	Rocket Launcher	PPS 040/2000	Fusion40	Magnum 40/4	G-Force 040/28	PPS 040/3000	Mercury	G-Force 040/28
AIBB-TEST												
Write Pixel	25,68 s	6,39	6,79	4,88	5,20	7,30	7,21	9,37	9,37	7,13	7,83	6,76
Sieve	63,60 s	18,22	20,06	13,77	17,67	18,17	16,91	22,88	20,00	10,00	18,17	18,17
Dhystone	1426 D/s	8,96	9,52	7,33	6,69	21,91	22,19	26,17	25,41	19,48	21,91	21,38
Sort	58,98 s	13,78	15,05	10,40	12,24	24,37	23,98	30,09	28,91	19,66	24,37	24,17
Matrix	22,10 s	17,27	18,73	13,00	15,35	22,55	21,25	29,08	26,31	14,54	22,55	22,55
IMath	95,64 s	34,04	37,65	27,17	29,89	47,82	48,80	57,61	56,26	42,70	47,82	47,82
MemTest	37,46 s	8,61	8,96	7,29	8,29	7,87	6,79	9,96	8,63	7,12	8,88	7,37
TGTest	11,82 s	2,17	2,19	2,13	2,14	2,38	2,41	2,38	2,21	2,31	2,38	2,40
Savage	702,54 s	415,70	459,18	415,70	418,18	169,70	164,92	191,95	23,16	150,12	169,70	162,63
FMmath	72,04 s	43,66	48,68	43,40	43,40	76,64	92,36	87,85	13,59	69,27	76,64	76,64
FMatrix	31,64 s	13,07	14,25	11,68	12,66	8,60	8,74	12,86	1,58	6,56	8,60	10,41
Beachball	1106,56 s	71,85	72,70	58,12	67,23	123,78	118,48	157,18	73,38	116,97	126,61	104,00
SWhetstone	24620 W/s	122,02	130,43	107,53	122,34	195,28	195,28	230,78	41,62	175,08	195,28	191,59
DWhetstone	27012 W/s	106,07	115,83	95,56	98,46	177,98	181,47	205,67	38,09	156,87	177,98	174,63
FTTrace	501,32 s	225,82	270,98	211,53	208,88	144,06	156,66	158,65	21,31	125,33	143,23	134,76
CplxTest	140,54 s	41,83	45,19	35,76	34,28	115,20	117,12	137,78	135,13	103,34	115,20	115,20
PRAXISTEST												
Imagine 2.0	24310 s	16,10	16,40	14,21	14,73	54,88	52,85	68,10	57,47	35,54	56,01	55,50
AdPro 2.0	445 s	10,35	10,85	8,24	8,90	18,54	16,48	23,42	20,23	10,35	20,23	18,54
Aquarium 1.15	296 s	8,46	8,71	6,88	7,22	14,10	14,10	15,58	12,33	10,57	15,58	14,80
Beckertext II	861 s	6,15	6,19	5,28	5,16	8,79	8,79	8,61	8,20	7,18	9,36	8,97
Lattice 5.0	765 s	11,95	12,14	7,50	6,43	9,11	15,94	18,66	13,91	8,41	17,00	17,00
Workbench 2.0	2866 s	8,01	8,19	5,65	6,30	9,55	12,46	16,86	8,68	11,02	12,46	15,49
DiskSpeed 4.1	372 KByte/s	1,84	1,85	1,45	1,66	1,98	1,30	1,68	1,52	1,52	1,68	1,60

liczby "prawdziwych" procesorów MC68030. Być może w kartach trafiliśmy na procesor w jakiejś wersji "pośredniej" i dlatego zachowywał się on, jakby nie miał tej jednostki. Wszystko to, co opisa- no powyżej, dotyczy starszej (25 MHz) wersji procesora. W związku z niezbyt ciekawymi doświadczeniami, związa- nymi z używaniem tego procesora w nowszej wersji (40 MHz) zrezygnowano z kopro- cesora zarządzającego pa- mięcią (PMMU). Mimo wszyst- ko, przy testach kart, które zawierały procesor MC68EC03 wymienialiśmy procesor na taką jego wersję, która nie zawierała PMMU.

Przy zakupie kart wyposa- żonych w ten procesor należy wziąć pod uwagę, że żaden producent takich kart nie gwarantuje prawidłowe- go działania koprocera zarządzającego pamięcią. W związku z tym możecie nie otrzymać gwarancji nawet na całą taką kartę. W przypadku kłopotów z działaniem funkcji systemu operacyjnego można skorzystać ze specjalizowa- nego oprogramowania, które "przepisuje" zawartość Kick- start ROM do 32-bitowej pa- mięci typu Fast, jednak w tym przypadku programy te muszą jednocześnie zapewnić zabezpieczenie tej ostatniej pamięci przed zapi-

sywaniem. UWAGA: Opro- gramowanie takie nie jest dołączone do wszystkich kart turbo.

Przejdźmy teraz do wyni- ków naszego testu:

Karty z procesorem MC68000: Tu zwycięzcą jest bez- sprzecznie Blizzard Board, który dzięki wyposażeniu go w dodatkową pamięć RAM oraz w pamięć typu Shadow wyraźnie zostawia w pobitym polu pozosta- łe przyspieszacze z tej grupy.

AdSpeed, który ma niewie- le gorsze osiągi — z powodu relatywnie wysokiej ce- ny (prawie taka sama jak Bliz- zard, choć nie ma "dodatko- wych" pamięci) nie mógł z nim konkurować.

Mach 2 (III miejsce) — w niektórych punktach jest nawet lepszy niż AdSpeed, jednak ocenę ogólną obniża jego gor- sza kompatybilność.

Neuroth 020 jest wprawdzie kartą z procesorem MC68020, jednak bez pamięci 32-bitowej w wielu testach osiągał gorsze wyniki niż karty z podwójnie taktowanym MC68000. Przy ocenie kart braliśmy pod uwa- gę głównie wyniki testu prakty- cznego.

Na tym przerywamy opis testu kart przyspieszających. Druga część — już za miesiąc.

AMIGA Magazin 8/92
Tłum. Marek Pampuch

EUREKA □ SOFT □ HARDWARE □

tel./fax (066)-362-072 (16h-20h) ul. Żwirki i Wigury 13a 62-300 WRZEŚNIA



Amiga Action Replay MK III (3.17)

Nowa moc dla twojej Amigi !!!
Doskonała pomoc dla każdego
programisty, hackera i gracza.

wersja do A500, plus, tylko 1550tys.

wersja do A2000 tylko 1720tys.

polska instrukcja, gwarancja.



Stacje dysków 3.5" i 5.25"

GOLDEN IMAGE lub 3-STATE przelotowe,
posiadają wyłącznik, stacje 5.25" posiadają
przełącznik 40/80 ścieżek. Odznaczają się cicha
pracą. cena stacji 3.5" 1150 tys.
cena stacji 5.25" 1300 tys.



Rozszerzenie 512 kb do A500 zegar, wyłącznik, na 4 kościach.

Golden Image cena 550 tys.

Ceny detaliczne, gwarancja, koszty przesyłki:
za pobraniem ok. 6% wartości.

Poza tym oferujemy:

Myszki "Fracky Mouse" 200 tys.
boczne rozszerzenia pamięci 2-8MB
dyski twarde 52MB, 105MB,
rozszerzenie pamięci do A500plus
osobne kontrolery, flicker fixery
oprogramowanie Publick Domain

Wyłącznie w sprzedaży hurtowej:

dyskietki 3.5", 5.25" DD, HD
No-name i firmowe również
na cele zaopatrzeniowo-inwesty-
cyjne (taniej).
podkładki pod myszy, dyskietki
czyszczące, BOXY na dyskietki
(najtaniej!).

Sklepy, hurtownie prosimy o kontakt. Szukamy dystrybutorów sprzętu, oraz zbytu na duże ilości dysków

PCKurier to informacyjny dwutygodnik (25 wydań rocznie) przeznaczony dla użytkowników komputerów osobistych. Składa się nań kilka bloków:

- ◆ Notes czyli zwięzłe notki o wydarzeniach, które miały miejsce oraz takich, które dopiero nastąpią;
- ◆ PCinfo czyli krótkie informacje o sprzęcie, oprogramowaniu i rynku mikrokomputerowym;
- ◆ PCmemo - rozbudowane informacje programów i sprzętu;
- ◆ znajdująca się zawsze na rozkładówce rubryka Pro memoria, w której publikowane są w formie zestawień, tabel itp. funkcje programów, porównania różnych kart, dysków itd., słowem informacje, które nawet jeśli nie są w danym momencie potrzebne, to warto zachować;
- ◆ Dla praktyków czyli rubryka z różnymi sztuczkami i rozwiązaniami najróżniejszych problemów;
- ◆ I wreszcie: Giełda czyli setki drobnych (gratisowych) ogłoszeń - Kupię, Sprzedam, Zamienię, Dam pracę, Szukam pracy.
- ◆ PCKurier ukazuje się od 1989 roku.
- ◆ Cena kioskowa: 7.000 zł.

◆ **W prenumeracie taniej: roczna (25 numerów) 155 tys. zł, półroczna (12 wydań) 80 tys. zł. Wysyłka pocztą gratis!**

Magazyn **AMIGA** to ilustrowany miesięcznik przeznaczony dla użytkowników komputerów Commodore Amiga — zarówno dla tych początkujących jak i dla zaawansowanych, zarówno dla interesujących się oprogramowaniem jak i tajnikami sprzętu. Część artykułów jest tłumaczeniem z najpopularniejszego na rynku niemieckim miesięcznika "AMIGA Magazin".

Wśród stałych rubryk Czytelnicy znajdują m.in.:

- ◆ AMIGA Play — opisy i oceny kilkunastu gier (nowości, ale także ulubionych "klasyków").
- ◆ Public Domain — opisy dyskietek najpopularniejszej biblioteki oprogramowania Public Domain — dyskietki Fisha.
- ◆ Kuferek AMIGI czyli Tips&Trics.
- ◆ Testy sprzętu i oprogramowania.
- ◆ Wszystkie te rzeczy znajdują Państwo na 80 barwnych stronach miesięcznika.
- ◆ Cena kioskowa: 20.000 zł.

◆ W prenumeracie:
za 6 numerów — 120.000 zł,
za 12 numerów — 240.000 zł.

◆ **Wysyłka pocztą gratis!**

DECforum to ilustrowany kwartalnik o objętości ok. 60 stron przeznaczony dla użytkowników systemów komputerowych firmy Digital Equipment oraz — nieco szerzej — dla użytkowników systemów mini i większych. Pismo wydawane jest na zlecenie i pod merytoryczną kontrolą Digital Equipment.

W piśmie pojawiają się między innymi następujące rubryki:

- ◆ Nowe idee
- ◆ Oprogramowanie
- ◆ Sprzęt
- ◆ Nowe produkty
- ◆ Cena detaliczna (DECforum można kupić w siedzibie wydawnictwa oraz w księgarniach technicznych): 20.000 zł.

◆ **W prenumeracie: za 4 numery — 80.000 zł. Wysyłka pocztą gratis!**

WYDAWNICTWO
LUPUS

Zasady prenumerowania czasopism w Wydawnictwa LUPUS

1. Prenumerata przyjmowana jest na taką liczbę numerów jaka została zaznaczona w tabeli na kuponie.
2. Prenumerata przyjmowana jest od najbliższego numeru po otrzymaniu kuponu przez Wydawnictwo.
3. Prenumeratę można opłacić także w siedzibie Wydawnictwa.
4. Wszelkie wątpliwości można wyjaśnić telefonicznie: (0-22)410031 w. 154.
5. Wydawnictwo nie ponosi odpowiedzialności za problemy wynikające z błędnego wypełnienia kuponu.

DECforum						
PCvirus						
CADforum						
AMIGA						
ENTER						
PCKurier						
						to moja pierwsza prenumerata

kupon ważny do 10.11.92
4 6 12 25

DECforum						
PCvirus						
CADforum						
AMIGA						
ENTER						
PCKurier						
						to moja pierwsza prenumerata

kupon ważny do 10.11.92
4 6 12 25

DECforum						
PCvirus						
CADforum						
AMIGA						
ENTER						
PCKurier						
						to moja pierwsza prenumerata

ENTER to ilustrowany, popularny, wysonakładowy miesięcznik poświęcony technice mikrokomputerowej i jej zastosowaniom. Magazyn ENTER adresowany jest do użytkowników różnych komputerów, w szczególności: Atari ST, Commodore Amiga, IBM PC, Macintosh. Także osoby nie posiadające komputera a zainteresowane tą techniką znajdują w miesięczniku wiele ciekawych materiałów. ENTER jest bogato ilustrowany i wydawany na wysokim poziomie edytorskim. Na szczególną uwagę zasługują trzy rubryki pisma:

- ♣ **RAPORT** - w każdym numerze publikowany jest test porównawczy sprzętu lub oprogramowania (np. drukarki, 386., skanery, arkusze kalkulacyjne) dający czytelnikowi wszechstronną wiedzę o oferowanych na rynku produktach;
- ♣ **LABORATORIUM** - nieodłączną częścią miesięcznika są testy sprzętu i oprogramowania publikowane w każdym numerze;
- ♣ **KONSyliUM** - rzecz w polskiej prasie komputerowej dotychczas nie spotykana czyli porady w formie pytań czytelników i związanych, precyzyjnych odpowiedzi ekspertów (kilka - kilkanaście pytań w jednym numerze).
- ♣ **Cena kioskowa: 17.000 zł**
- ♣ **W prenumeracie taniej: za 6 numerów 95.000 zł, za 12 numerów 180.000 zł, wysyłka pocztą gratis!**

CADforum to dwumiesięcznik (6 wydań rocznie) przeznaczony dla osób zainteresowanych komputerowym wspomaganie projektowania (CAD czyli Computer Aided Design). W piśmie przedstawione są różne systemy CAD - m. in. AutoCAD, LogoCAD, SysCAD... Różne także obszary zastosowań leżą w kręgu zainteresowania pisma: architektura, budownictwo, geodezja, kartografia, mechanika, elektronika i projektowanie obwodów, grafika itd. Wiele jest informacji praktycznych, nadających się do natychmiastowego wykorzystania (m. in. programy w LISP).

- ♥ CADforum jest pismem fachowym. Mimo tego jednak pismo adresowane jest nie tylko do osób profesjonalnie zajmujących się CAD, ale także do wszystkich tych, którzy chcą (choćby wstępnie) poznać temat, dowiedzieć się jakie w interesujących ich dziedzinach istnieją możliwości stosowania techniki komputerowej. Projektowanie bez komputera to dzisiaj już archaizm.
- ♥ Pismo jest jedynym tego typu wydawnictwem w Polsce (istnieje od 1989 roku).
- ♥ **Cena detaliczna (CADforum dostępny jest w księgarniach technicznych): 21.000 zł.**
- ♥ **W prenumeracie taniej: 100.000 zł za 6 numerów.**

PCvirus to wydawany w formie biuletynu dyskietkowego dwumiesięcznik poświęcony wirusom komputerowym i walce z nimi. PCvirus wydaje najmocniejszy zespół jaki można sobie w naszym kraju wyobrazić. Tworzą go: Andrzej Kadłof (twórca programu antywirusowego PAW) oraz Marek Sell (twórca programu antywirusowego Mks_VIR). Nikt, tak jak oni, nie zna tej problematyki. Na dyskietkach kolejnych numerów znajduje się m. in. unikatowa baza danych wszystkich dotychczas schwytych wirusów zawierająca komplet danych pozwalających na identyfikację wirusa i stworzenie własnej szczepionki. Rozprowadzane są także najnowsze wersje pakietu antywirusowego firmy McAfee.

- ♣ Poza tymi "rarytasami" czytelnicy znajdą wyczerpujący serwis informacyjny na temat wirusów komputerowych, zasady profilaktyki, porady itd.
- ♣ PCvirus jest pismem całkowicie unikatowym i to zarówno ze względu na formę (dyskietki) jak i treść.
- ♣ **Pismo można kupić jedynie w siedzibie wydawnictwa. W prenumeracie taniej: 180.000 zł za 6 numerów.**

WYDAWNICTWO LUPUS

Jak zaprenumerować czasopismo Wydawnictwa LUPUS?

1. Podjąć decyzję, które z czasopism chce się prenumerować.
2. Wypełnić starannie (najlepiej drukowanym piśmem) wszystkie odniki zamieszczonego obok kuponu.
3. Na odwrocie zaznaczyć krzyżykami, które z czasopism prenumerujemy, ile numerów oraz czy dokonujemy prenumeraty po raz pierwszy.
4. Wyjąć kupon i korzystając z niego dokonać wpłaty na pocztę lub w banku.
5. To wszystko.

Pokwitowanie dla Wpłacającego

zł
słownie
wpłacający

adres

na rachunek:
LUPUS Sp. z o. o.
Warszawa, ul. Stępińska 22/30

IX Oddział PKO BP w Warszawie
r-k. nr. 1599-318121-136

Oddinek dla Posiadacza r-ku

zł
słownie
wpłacający

adres

na rachunek:
LUPUS Sp. z o. o.
Warszawa, ul. Stępińska 22/30

IX Oddział PKO BP w Warszawie
r-k. nr. 1599-318121-136

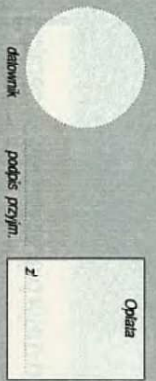
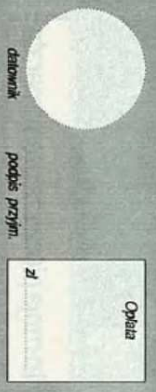
Oddinek dla Banku

zł
słownie
wpłacający

adres

na rachunek:
LUPUS Sp. z o. o.
Warszawa, ul. Stępińska 22/30

IX Oddział PKO BP w Warszawie
r-k. nr. 1599-318121-136





Wojciech Bruszewski jest profesorem w Państwowej Wyższej Szkole Sztuk Plastycznych w Poznaniu, gdzie prowadzi Pracownię Grafiki Komputerowej. Jego studenci pracują na Amigach. Dziś inaugurujemy cykl wykładów pod wspólnym hasłem mikro-KINO.

Wojciech Bruszewski

Mikro-Kino będzie miejscem spotkań słabo lub średnio zaawansowanych miłośników Amigi. Zakładam, że będą to głównie młodzi ludzie, którzy rozbili już kilkadziesiąt sportowych samochodów, zestrzelili kilkadziesiąt radzieckich Migów, uszkodzili co najmniej jeden joystick i zaczynają przejawiać zainteresowanie własną mikrotwórczością.

Myślę również, że wdzięcznymi widzami mikro-Kina będą nauczyciele, którzy mają ambicję wprowadzać Amigę do szkół jako komputer edukacyjny. Odczują oni brak oprogramowania dydaktycznego w ojczystym języku. Być może niektórzy z nich będą wypełniać tę programową dziurę, realizując własne mikrofilmy.

W mikro-Kinie będzie się mówić na temat twórczości o charakterze wizualnym i akustycznym; o grafice i animacji; o syntezie mowy i dźwiękach naturalnych; o łączeniu obrazów i dźwięków w komputerowe kino z silnym akcentem na interakcję widza.

Mikro-Kino ma być w założeniu inspiracją twórczą i treningiem, który zmierza do swobodnego kształtowania własnej wizji na multimedialnej maszynie, jaką jest Amiga.

WARSZTAT

Do realizacji filmów nie będziemy angażować urządzeń drogich lub wysoce specjalistycznych.

Naszą podstawową narzędziownią będą:

1. Amiga
2. Videodigitizer
3. Audiodigitizer
4. DPaint
5. The Director

AMIGA

Absolutnym minimum jest Amiga 500 z rozszerzeniem pamięci do 1 MB. W pamięci Amigi są dwa obszary nazywane

CHIP RAM i FAST RAM. Generalnie rzecz biorąc — im więcej CHIP RAM, tym swobodniej manipulujemy obrazami i dźwiękami.

W najbardziej popularnej A-500, po rozszerzeniu pamięci do 1 MB, mamy do czynienia z połową CHIP i połową FAST RAM. Z użyciem lutownicy, a w starych Amigach z koniecznością wymiany chipu o nazwie FatAgnus, można skonfigurować pamięć jako 1 MB CHIP RAM. Ale uwaga: DPaint nie działa bez FAST RAM-u.

W A-3000, A-500+, A-600 jest możliwość ustawienia 2MB CHIP RAM.

Minimum komfortu to dwie stacje dysków. Nieustanne wyjmowanie i wkładanie dyskietek do jednej stacji może zmniejszyć do pracy.

Prawdziwy komfort to twardy dysk.

VIDEODIGITIZER

czyli komputerowa "kamera reprodukcyjna". Pozwala wprowadzić do komputerowego programu obrazy natury.

Nam wystarczy jeden z najprostszyc modeli: Digi View (New Tek) lub Deluxe View (Hegenau Computer), lub inny.

AUDIODIGITIZER

czyli komputerowy "magnetofon". Pozwala wprowadzać do programów muzycznych dźwięki naturalne.

Może to być Audio Master (Aegis), Deluxe Sound (Hegenau Computer), Future Sound (Applied Visions, Inc.) lub inny.

DPaint (Electronic Arts)

czyli "dobra farba". Najbardziej zaawansowany i jednocześnie najbardziej popularny program malarski. Pozwala uprawiać nie tylko elektroniczne malarstwo, ale również komputerowy film animowany.

Może być DPaint III lub DPaint IV.

THE DIRECTOR (Right Answers)

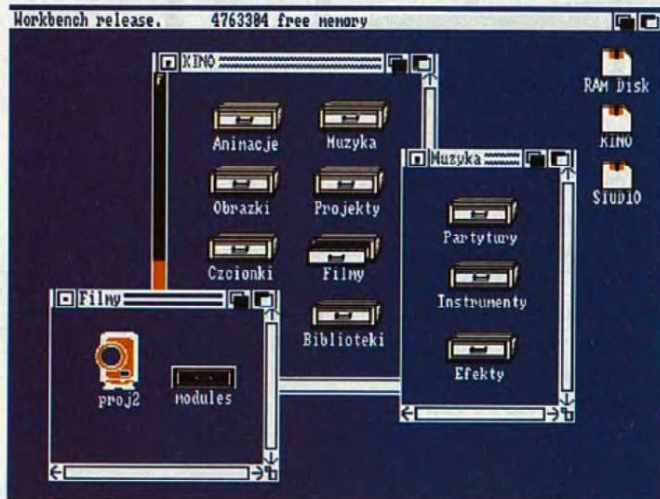
czyli "reżyser"; a dokładniej, mikrojęzyk zorientowany na multimedia. Wszystkie filmy będziemy robić z zastosowaniem wersji 2.0. Użytkownicy wersji 1.2/1.3 powinni zatroszczyć się o up-date.

Najlepiej u samego źródła:

Right Answers,
Box 3699, Torrance, CA 90510, USA

DYSKIETKI ROBOCZE

Do pracy należy przygotować trzy dyskietki, z których jedna o nazwie STUDIO, będzie okrojonym Workbenchem wraz z narzędziami; druga, o nazwie STUDIO-2, pomieści resztę narzędzi; a trzecia — KINO, przeznaczona będzie na materiały do produkcji filmów i gotowe filmy.



Dyskietka STUDIO.

STUDIO

Wykonaj kopię oryginalnego Workbencha. Na kopii (i tylko na kopii) skasuj całe szuflady Utilities, Prefs, Expansion.

Szufladę Empty skopiuj do RAM dysku.

Szufladzie Empty na dyskietce STUDIO zmień nazwę na Director.

W szufladzie C niech pozostaną:

AddBuffers	Assign	Binddrivers
Break	CD	Copy
Date	Delete	Dir
Echo	EndCLI	Execute
Failat	FF	IconX
List	LoadWB	Makedir
Mount	NewCLI	NewShell
Path	Resident	Run
SetClock	SetPatch	Sort
Stack	Type	Wait

W szufladzie DEVS/KEYMAPS pozostaw tylko jedną key-map. Powinna to być spolszczona mapa klawiatury.

W szufladzie DEVS/PRINTERS pozostaw tylko jeden driver drukarki.

W szufladzie FONTS pozostaw tylko Topaz. Do szuflady FONTS skopiuj spolszczony Topaz (na przykład TopPL) i czcionkę DPaint, jeżeli pracować będziesz programem DPaint w wersji IV. W wersji III czcionka ta jest niepotrzebna.

W szufladzie S pozostaw tylko: Startup-sequence i Startup II.

Otwórz edytor tekstów, na przykład CED lub DEdit, i do startup-sequence dopisz dwie nowe linie:

```
assign director: Studio:director
```

```
assign mod: Studio:director/modules
```

W lini Path ... zlikwiduj słowa Sys:prefs Sys:Utilities.
Jeżeli w miejsce Topazu instalujesz polską czcionkę, na przykład TopPL, zmodyfikuj linię FF ...
FF Studio:fonts/TopPL.font
Instalując polską mapę klawiatury, zmodyfikuj też linię SetMap ...
Studio:System/SetMap PLD
W szufladzie SYSTEM pozostaw tylko:
SetMap FastMemFirst Format DiskCopy
Na tak przygotowany dysk, do szuflady Director, skopiuj program The Director.

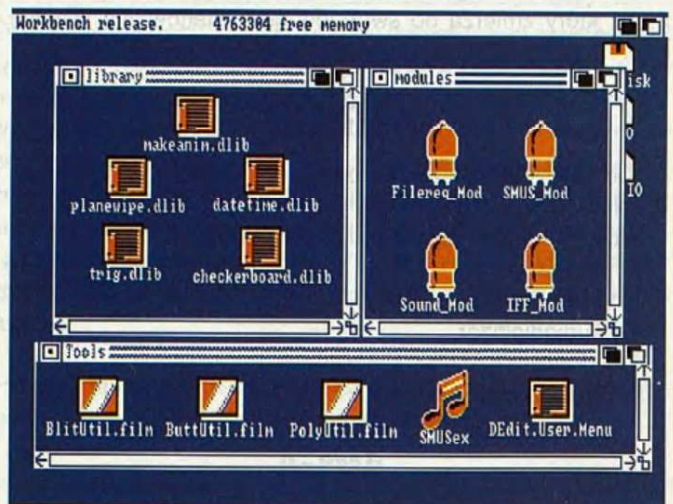


Dyskietka STUDIO. Narzędzia, moduły i biblioteki Directora.

Najważniejsze części pakietu The Director:

- dEdit — edytor tekstu do pisania "scenariuszy" filmów;
- d2 — The Director w wersji 2; realizuje scenariusze filmów;
- proj2 — projektor, który "wyświetla" filmy;
- Sound_mod — moduł, który odtwarza dźwięki naturalne;
- Smus_mod — moduł, który gra partytury muzyczne SMUS;

012



Dyskietka STUDIO-2.



Niestety, na dysku STUDIO nie zmieszczą się pozostałe narzędzia, chyba że STUDIO jest dyskiem twardym.

Pozostałe narzędzia mogą być zgromadzone na innym dysku bez elementów Workbencha. Na takim "gołym" dysku, nazwanym na przykład STUDIO-2, DPaint, videodigitizer i audiodigitizer oraz kilka innych programów zmieści się na pewno.

Dyskietki STUDIO należy zabezpieczyć przed zapisem (okienko otwarte) i wykonać kopie bezpieczeństwa. Awaryjne dyski zdarzają się często.

KINO

Na materiały wizualne, materiały akustyczne, teksty i filmy należy przeznaczyć czystą sformatowaną dyskietkę. Nadać jej nazwę KINO.



Dyskietka KINO.

W RAM dysku mamy pustą szufladę Empty, skopiowaną tam przez proste przeciągnięcie ikony z okna Workbencha do okna RAM dysku.

Zmień nazwę Empty na Animacje. Skopiuj szufladę na dyskietkę KINO.

W RAM dysku zmień nazwę Szuflady na Obrazki. Skopiuj na dyskietkę KINO.

I tak dalej, ...

Przeznaczenie szuflad jest następujące:

Animacje animacje zapisane jako AnimFile;

Obrazki obrazy graficzne jako IFF;

Czcionki, czyli fonts;

Projekty teksty i skrypty programów;

Biblioteki własne narzędzia programowe;

Filmy skompilowane skrypty;

Muzyka, a w niej:

Partytury, czyli scores zapisane jako SMUS;

Instrumenty takie jak w Sonixie;

Efekty, czyli dźwięki naturalne tzw. dumps.

Zwróć uwagę, że we wszystkich polskich nazwach nie ma polskich znaków, takich jak ą,ć,ę,ł,ń,ó,ś,ż. Jestem zdania, że w nazwach, z którymi ma do czynienia AmigaDOS bezpieczniej jest unikać znaków o adresach wyższych niż 127.

Ponadto w nazwach nie używamy: spacji [], dwukropka [:], cudzysłowu ["] i znaku slash [/].

Do szuflady Filmy skopiuj Proj2 i całą szufladę Modules z dyskietki STUDIO. W ten sposób micro-filmy mogą być wyświetlane niezależnie od dyskietki STUDIO. Ale tu uwaga! Director umieszcza gotowe filmy w RAM dysku. W ikonie zapisana jest informacja, gdzie szukać programu Projektor. Kopiując film do szuflady Filmy, należy zmienić tę informację.

Jeden klik w ikonę filmu, prawym klawiszem myszy wybrać INFO z menu Workbencha. W linii DEFAULT TOOL napisać Proj2 i kliknąć w SAVE.

Dyskietka KINO staje się kompletnym dyskiem, który może być przeznaczony do dystrybucji. Projector i moduły Director są programami Public Domain.

Warsztat pracy jest gotów. Być może było to nudne, jak ostrzenie ołówków, ale ostrzenia ołówków nie da się uniknąć.

Na koniec uwaga w kierunku użytkowników pirackiego oprogramowania.

Nowoczesne programy bardzo często mają konstrukcję modułową. Nie wystarczy przeciągnąć ikonę z okna do okna. Nie wszystkie moduły muszą mieć ikony. Aby program działał — musi być kompletny.

Takim programem jest The Director.



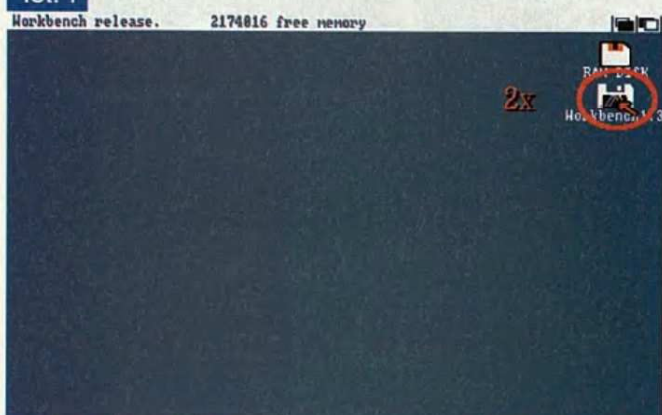
ITY TO POTRAFISZ

W dzisiejszym odcinku obrazków dla początkujących zajmiemy się oknami

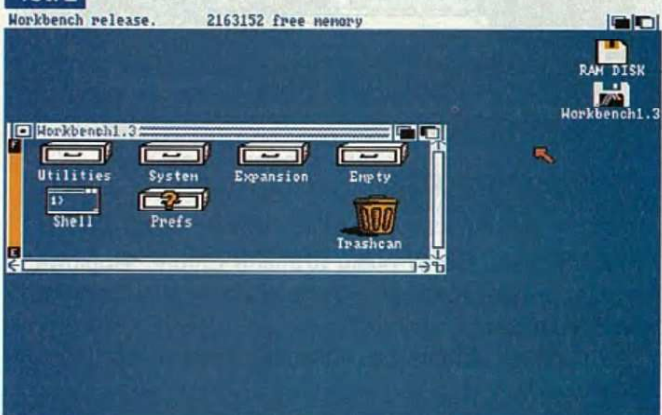
- 1 Po załadowaniu Workbencha na ekranie pojawia się taki obraz.
- 2 Ustawmy wskaźnik myszki na rysunek dyskietki i dwukrotnie naciśnijmy lewy klawisz.
- 3 Pojawi się ramka zwana OKNEM.
- 4 Popatrzymy na listwę tytułową. Mamy tam między innymi nazwę okna.
- 5 Jeśli nazwa ta jest niewyraźna, wówczas z tym oknem nie można pracować.
- 6 Okno takie należy "uaktywnić". Ustawmy wskaźnik w dowolnym miejscu wewnątrz okienka (byle nie na rysunkach zwanych IKONAMI) i naciśnijmy raz lewy klawisz myszki.
- 7 Na brzegach okna znajdują się tzw. gadżety. Jeśli ustawisz wskaźnik na ten gadżet i naciśniesz dwa razy lewy klawisz myszki, okno zniknie z ekranu. Aby je "odzyskać" — postąp jak w punkcie 2.
- 8 Aby przesunąć okno w inne miejsce ekranu — ustaw wskaźnik na listwie tytułowej, naciśnij lewy klawisz myszki i przytrzymując go, poruszaj myszką. Po uzyskaniu właściwej pozycji puść klawisz.
- 9 Chcąc zmienić wielkość okna — ustaw wskaźnik na ten gadżet, naciśnij lewy klawisz myszki i przytrzymując go, przesunij ją w miejsce, gdzie ma się znaleźć prawy dolny róg okna. Po ustawieniu — puść klawisz.
- 10 Jeśli biała kreska na dole lub po prawej stronie nie zajmuje całej ramki, możesz "przesunąć" zawartość okna. Ustaw wskaźnik na jednej z tych kresek i przytrzymując naciśnięty klawisz myszki, przesunij "zwierzaka" w kierunku "pustego" niebieskiego miejsca w ramce. Po przesunięciu — puść klawisz.
- 11 Jeśli na ekranie jest kilka zachodzących na siebie okien, możesz je wysuwać i chować, wybierając myszką jeden z tych dwóch gadżetów.
- 12 Aby uruchomić program, ustaw wskaźnik na ikonę podpisaną nazwą programu i dwa razy naciśnij lewy klawisz myszki.



fot. 1



fot. 2



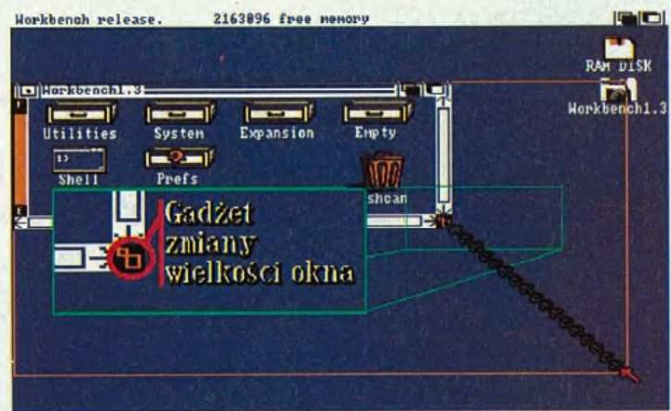
fot.3



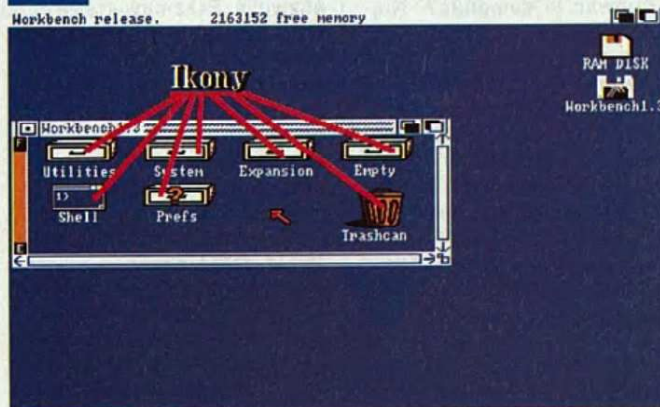
fot.4



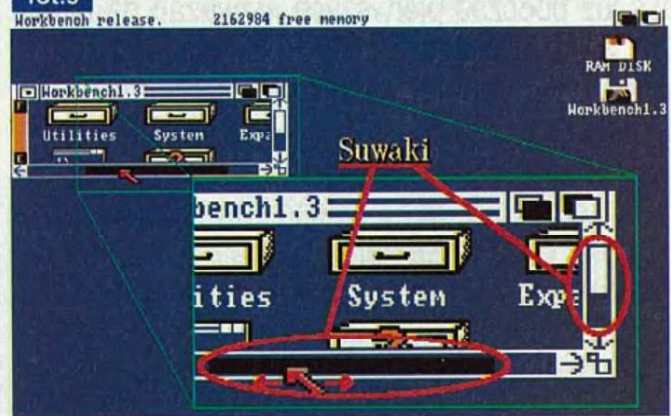
fot.5



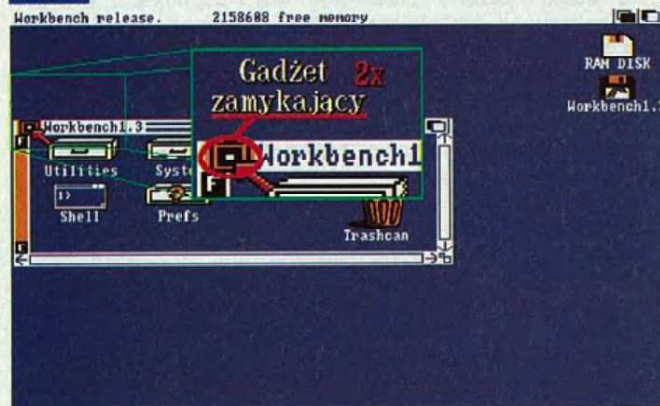
fot.9



fot.6



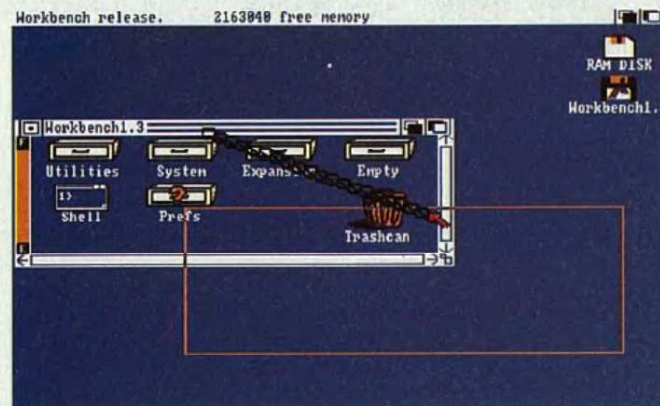
fot.10



fot.7



fot.11



fot.8



fot.12

Amiga i IBM (cz. 1)

DO ŁEZKI ŁEZKA...

...aż będę niebieska. Refren piosenki autorstwa nieodżałowanego Jonasza Kofty przyplątał się do nas już podczas pierwszych rozważań na temat "jak z Amigi zrobić IBM?". Rozpoczynamy serię artykułów (zawierających zarówno tłumaczenia z AMIGA Magazin, jak i materiały uzyskane na zamówienie naszej redakcji) dotyczących tego tematu. Właściwie poza opisami programów i kart emulujących PC ten "temat-rzeka" nie był nigdzie dotąd w polskiej prasie komputerowej poruszany.

Marek Pampuch, Rafał Wiosna

Początkowo większość firm komputerowych manifestowała swoją niezależność od "wielkiego niebieskiego brata". Niestety nie miały takiej siły przebicia, jak IBMowski moloch, który — widząc że jego opracowany naprędce mikrokomputer ustępuje znacznie wyrobom zarówno Apple, jak i Commodore, opartym na znacznie lepszym procesorze niż przestarzały Intel — wykonał mistrzowskie pociągnięcie. Wprawdzie "odtajnienie" szczegółów konstrukcyjnych początkowo spowodowało lekkie obniżenie obrotów, ale na dłuższą metę okazało się pomysłem genialnym. I w taki sposób, niestety, świat został zalany czymś wielkim i niebieskim. Dwie wymienione firmy "opierały" się najdłużej, ale w końcu też się poddały. Ma to swoje odbicie i w świecie oprogramowania. Liczba dostępnych programów na poszczególne komputery jest zazwyczaj proporcjonalna do liczby sprzedanych egzemplarzy. Nawet jeśli ktoś wyprodukuje "porządny" program na inny komputer — to

zdając sobie sprawę z niewielkiego zbytu — odpowiednio go wyceni. Wynika z tego to, że cena z kolei jest odwrotnie proporcjonalna do liczby komputerów na rynku. Czyli: im więcej sprzedanych komputerów — tym więcej tanich programów. Dział tu sprzężenie zwrotne. Im więcej programów, tym więcej kupuje się komputerów. Po przykrych doświadczeniach z ciągle psującymi się "tajwanami" następnym razem prawie każdy kupi "oryginalnego" IBM-a. Jak jednak ma się to wszystko do naszej Amigi?

Firmy Commodore i Apple zaczęły, podobnie jak inne firmy, produkować IBMowskie klony — nie zaprzestały jednak budowy i ulepszania swych "flagowców" — Macintosha i Amigi. W ramach ulepszania — spróbowano także tak "dopasować" te komputery, aby nie tracąc niczego ze swych cech, mogły podawać "peceta". Prędzej czy później każdy Amigant zetknie się z tą tematyką. Po okresie fascynacji grami, po wykorzystaniu kilkunastu programów użytkownik Amigi staje przed nie lada problemem. W jaki sposób spowodować, aby nasz ukochany komputer był chociaż w nie-

wielkim stopniu zgodny z "wielkim niebieskim bratem"? I czy w ogóle warto próbować to osiągnąć? Najczęściej pytania te stawiamy sobie w okolicznościach nad wyraz nieprzyjemnych. Przypuśćmy, że napisałeś wspinały artykuł, niekoniecznie na temat Amigi, wykorzystując w tym celu swoją Amigę. Wysyłasz dyskietkę do redakcji gazety, która chciałaby ten tekst opublikować. Niestety w redakcji rozpanoszył się brutalnie IBM, zaś dział składu nie potrafi "przełożyć" formatu Amigi na aktualnie stosowany przez nich "standard" IBM-a. Takie same rozterki przeżyjesz w sytuacji, gdy chcesz przyspieszyć wykonywanie zadań służbowych za pomocą — przypuśćmy — programu inżynierskiego. Niestety, pokłóciłeś się z szefem ETO, tak że w kolejce do jedyne go w biurze IBM-a jesteś zawsze ostatni, i to przeważnie już po terminie oddania projektu. Może też się zdarzyć, że jesteś studentem informatyki, tuż przed zaliczeniem. Cóż z tego, że napisałeś program w Aztec C, o niebo lepszy niż programy reszty twoich kolegów z grupy. Niestety, asystent ma IBM-a i żaden inny komputer dla niego nie istnieje. Jakże złowieszco brzmią słowa: "Jeśli nie przyniesie pan jutro tego programu w Turbo Pascalu — nie może pan nawet marzyć o zaliczeniu". Czy to Twoja wina, że asystent nigdy nie pojmie faktu, iż Intel jest tysiąc razy gorszy od Motoroli albo że firma Commodore nie może dogadać się z Borlandem? Z facetem, który ma na oczach kłapki z napisem IBM — nie dogadasz się, a akurat okoliczności sprawiają, że jest panem Twojego losu.

Idealem byłoby mieć oba komputery, ale na to rozwiązanie niewiele osób może sobie pozwolić. Przemawiają za nim zapewne także drwiny znanych posiadaczy IBM-a, twierdzących z przekąsem, że jeśli Amiga nadaje się do czegokolwiek, to tylko do gier. Pożytecznie także byłoby rozszerzyć posiadany zbiór programów na Amigę o takie, które wprawdzie istnieją, lecz żadna firma handlująca programami nie odważy się ich sprowadzić do Polski, zaś tzw. piraci zajmują się wyłącznie grami. Tłucze chandra... Nachodzą czarne myśli: "A może by tak sprzedać Amisję i kupić PC?" Nie rób tego. Z naszą niewielką pomocą dasz sobie radę z prawie wszystkimi swoimi problemami dotyczącymi IBM.

W naszym cyklu artykułów postaramy się pomóc zarówno tym, których nie stać na zakup drogiego dodatkowego sprzętu, jak i tym, którzy zastanawiają się, jaką przystawkę wybrać, a także tym, którzy już się na jakąś przystawkę zdecydowali, lecz mają kłopoty z jej obsługą.

Analizując sposoby emulacji IBM-a na Amidze, można wyróżnić trzy zasadnicze jej rodzaje:

- emulacja pełna — pozwalająca na "przekształcenie" Amigi w taki sposób, aby działała tak, jak zwykły IBM. Emulacja taka zazwyczaj wykonywana jest za pomocą dodatkowego sprzętu. Przykładem może być tu karta emulująca produkowana przez firmę KCS.

- emulacja częściowa — pozwalająca na wykonywanie większości czynności IBM-a na Amidze. Robi się to za pomocą programu lub dodatkowego sprzętu, a czasami wykorzystuje oba te sposoby na raz. Przykładem może być

karta BridgeBoard emulująca IBM XT z 512 KB pamięci i taktowaniem 4,77 MHz w trybie graficznym CGA.

- emulacja minimalna — pozwalająca na zapisanie danych Amigi w takim formacie, który jest zrozumiały dla "wielkiego niebieskiego brata". Ponieważ w tej chwili praktycznie każdy ma w jakimś stopniu dostęp do "oryginalnego" IBM-a, ta najtańsza, programowa emulacja wystarczająca do realizacji większości naszych celów — jest najbardziej popularna. Przykładami mogą być programy DOS-2-DOS oraz CrossDOS (przenoszenie plików), a także Image Link czy Art Department Pro (zmiana formatu grafiki).

W naszym cyklu pokażemy Wam, jak najlepiej wykorzystać możliwości programów do "emulacji minimalnej", jak zwiększyć zakres "emulacji częściowej", a także przedstawimy wszystkie aktualnie dostępne karty emulujące. Zaczniemy oczywiście od emulacji dostępnej dla każdego.

TRANSFORMER

Pierwszy program, który miał emulować IBM-a, jest niezbyt udaną przeróbką oprogramowania instalującego kartę Sidecar (będzie opisano dalej). Program miał robić to samo, ale bez dodatkowego sprzętu. Efekt jest gorzej niż tragiczny. IBM, owszem — zostanie zaemulowany, ale:

- do dyspozycji masz jedynie 371 KB pamięci RAM, zaś

nawet gdy masz zainstalowane dodatkowe rozszerzenie pamięci (na przykład do 1 MB), program i tak nie widzi więcej niż 498 KB. (Przy 2 MB można uzyskać pełne 640 KB).

- W programach, które nawet w tak niewielkiej ilości pamięci powinny działać bez problemu (na przykład Norton Utilities 3.0) nie "chodzą" wszystkie opcje, inne (dBase, Wordstar) działają prawidłowo, ale za to z szybkością, przy której żółw wydaje się superekspresem, bowiem:

— Emulowany IBM ma taktowanie trudne do zmierzenia jakimkolwiek testem (prawdopodobnie jest to 0,0000000001 MHz).

— Występują problemy z przeczytaniem dyskietek sformatowanych na 720 KB, ale na "normalnym" PC. Jeżeli chcesz korzystać z danych z IBM-a, powinienes je przegrać na dyskietce sformatowanej wcześniej "pod emulatorem" na Amidze.

— Mimo "teoretycznej" możliwości współpracy z twardego dyskiem — podobno praktycznie nawet najlepszy kontroler GVP-II z dyskiem Seagate nie pracuje z tym programem.

— Jedynym "emulowanym" trybem graficznym jest najstarszy tryb MDA, wyłącznie tekstowy. Z tego powodu nie można uruchomić programów pracujących w trybie graficznym, jak na przykład Chiwriter. Nawet w trybie tekstowym obsługa ekranu jest przerażliwie wolna. Spróbujcie przykładowo napisać "program" w GWBASIC (działa,

ale bez grafiki) złożony z jednego wiersza:

```
PRINT "TRANSFORMER  
JEST PROGRAMEM EMULUJĄ-  
CYM PECETA"
```

uruchomcie ten program i zmierzcie czas jego wykonania.

— W momencie próby narysowania czegokolwiek, w najlepszym przypadku nic się nie stanie, w najgorszym — Amiga zawiesi się, a ponowne wczytanie emulatora i DOS — to około trzech minut (nie mówiąc o możliwej utracie danych).

Dziękuję, wystarczy... Nie zamierzamy się dłużej pastwić nad programem, o którym napisano już setki stron, i prawie wszystkie w tonacji niezbyt pochlebnej.

IBEM i PC-TASK

Są to emulatory programowe nowej generacji — oczywiście w porównaniu z Transformerem. Oba potrafią emulować kartę CGA, oba otwierają swój własny ekran i pozwalają na pracę równoległą z aplikacjami Amigowskimi. Niestety, ten pierwszy dotarł do nas na razie w wersji demo i nie możemy autorytatywnie określić jego możliwości. Natomiast PC-Task nie jest "demo" i dzięki temu użytkownik może "zaemulować" sobie PC/XT 1 MHz... Niestety, na "gołej" A500 nic więcej nie da się uzyskać. Testy przeprowadzone na A3000 wskazują na to, że częstotliwość zegara wymaganego PC wzrasta wraz z mocą komputera. Tak

więc szczęśliwcy z A3000 lub A2000 z odpowiednią kartą przyspieszającą (najlepiej 68040 50 MHz) mogą w pełni oceniać walory programu. Nam pozostaje jednak prawie minutowe czekanie na wgranie Norton Commandera (z dyskietki). Przy okazji można wspomnieć, że PC-Task jest w stanie wykorzystać dołączony twardy dysk i zrobić na nim pseudo-partycję.

Od emulatorów sprzętowych przeszliśmy do programów wspomagających.

DOS-2-DOS

Po smutnych doświadczeniach z Transformerem zapewne rozplyną się marzenia o programowej emulacji IBM-a. Spróbujmy zatem emulacji minimalnej. Klasykiem jest tu program DOS-2-DOS. Nie jest to w pełnym tego słowa znaczeniu program emulujący, lecz program kopiujący, który może jednocześnie przekształcać format pomiędzy Amigowskim, IBM-owskim i Atari. Zgodnie z tematyką artykułu zajmijmy się na razie tymi pierwszymi dwoma.

Po uruchomieniu programu ukaże się na ekranie napis:

```
Enter the device to be used  
for MS-DOS/ATARI files: (po-  
daj nazwę urządzenia,  
jakiego chcesz używać do  
zbiorów MS-DOS lub Atari)
```

Wpisujemy tutaj symbol stacji dysków, z dyskietką w formacie odpowiednim dla IBM, na jaką (lub z jakiej) chcemy skopiować interesujące nas zbiory. Może to być stacja wewnętrzna DF0.

```
Amiga Transformer
Copyright 1985, 1986, 1987, Smile Research
Version 1.21
Available memory: 640K

Drive A: is of 8. B: is off.

Please insert a DOS diskette in drive A
press RETURN to start the DOS
or
press ESC to return to the Amiga DOS
```

```

Type MCL or ? for summary of Dos-2-Dos commands.

D2D> dir A1:

Directory of A1:\
MCL14.LMA      19745  8-14-92 11:35p
               1 file(s),    789632 bytes free

D2D> ?

Dos-2-Dos Command Summary:

Display directory..... DIR or DIR drive:path
Change current directory... CD drive:path
Display MCL file contents... type drive:path/file
Copy a file (general form)... copy drive:path/file drive:path/file -A -R
Copy one MCL file..... copy MCL:file.MCL drive:path/file -A -R
Copy MS-DOS/Atari files..... copy MCL:file.MCL
Copy AmigaDOS files..... copy MCL:file.MCL
Copy and convert MCL files... add -C to COPY command line
Suppress "Replace" question... add -S to COPY command line
Delete a file..... DEL drive:path/file
Format an MS-DOS hard disk... format /A
Format an Atari 512KB disk... format /A
Display this summary..... type ?
Select another MS-DOS drive... dir A:
Exit to AmigaDOS..... exit, or X

D2D> ||
```

Jednakże po jej wybraniu będzie ona czytała wyłącznie dyski PC, wobec czego posiadacze jednej stacji są zmuszeni do skopiowania plików, które mają być najpierw przeniesione do ram-dysku. Wszystkie inne zainstalowane stacje dysków są przez program akceptowane.

Po wprowadzeniu nazwy stacji i naciśnięciu [Return] pojawi się kolejny komunikat:

Type HELP or ? for summary of Dos-2-Dos commands (wpisz HELP lub ? aby uzyskać "streszczenie" rozkazów programu)

Ponieważ z reguły jesteśmy ludźmi ciekawymi, a nie mamy akurat czasu na przeczytanie instrukcji, zaś ściągę są pożyteczne — zróbmy to, o co prosi nas program.

W tym momencie na ekranie pojawi się spis rozkazów programu i ich składnia. Nie wątpimy w to, że podczas pracy z Amigą zdążyliście nauczyć się angielskiego, a zatem nie będziemy tłumaczyć zawartości ściągę. Jak widać, program umożliwia następujące czynności: wyświetlenie katalogu, zmiana katalogu aktualnego, wyświetlenie zawartości zbioru ASCII, kopiowanie zbiorów z konwersją i z "zastępowaniem" poprzednio nagranych wersji, usuwanie zbiorów, formatowanie dysków w formacie MS-DOS (720 KB) oraz Atari i zmianę dysku, na którym chcemy przechowywać dane MS-DOS/Atari, a także, rzecz jasna, zakończenie pracy programu i powrót do Amiga DOS. UWAGA: Program nie potrafi sformatować dyskietki MS-DOS, nagrywając przy tym system.

W celu przeniesienia zbioru z Amigi na dysk PC, należy wydać polecenie COPY PRZYKLAD.TXT DfX:, gdzie PRZYKLAD.TXT to plik do przeniesienia, a DfX: to stacja dysków zadeklarowana na początku jako czytająca dyski IBM (oczywiście należy napisać na przykład Df1:, a nie DfX:!!). W drugą stronę jest podobnie: COPY DfX:PECETOWY.TXT RAM:, jeżeli chcemy przenieść plik z PC do ram-dysku. Jeśli chcemy stosować tzw. wildcards, czyli kryteria wyboru plików, to DOS-2-DOS kryje w sobie

pewną niespodziankę. Otóż przy odwoływaniu się do urządzeń Amigi należy stosować amigowski zapis kryteriów, a przy wybranej stacji PC — kryteria wzięte z IBM. Najlepiej wyjaśnić to na przykładach:

COPY RAM:#? DfX: — skopiowanie wszystkich plików z ram-dysku na dysk PC,

COPY DH0:TEKSTY/#? MAZ DfX: — skopiowanie wszystkich plików, których nazwa kończy się sekwencją ".MAZ" z katalogu DH0: na dysk PC,

COPY Df1:* EXE RAM: — skopiowanie wszystkich plików z rozszerzeniem EXE z dysku PC do ram-dysku Amigi.

Jak widać, w Amidze sekwencja #? zastępuje dowolne znaki, natomiast na PC czyni to kryterium *.*.

CROSSDOS

Jest to nie mniej popularny program pozwalający na wykorzystywanie dysków PC w Amidze niż DOS-2-DOS. CrossDOS ma tę zaletę, że "podłącza" urządzenia logiczne Dlx: (odpowiadające stacji DfX:) pozwalające czytać i zapisywać na dyskach PC-towych tak jak na Amigowskich. Jest to bardzo wygodne rozwiązanie, gdyż nie trzeba używać "topornego" interfejsu użytkownika zastosowanego w DOS-2-DOS - wystarczy DiskMaster czy Opus Directory. Teoretycznie możliwe jest korzystanie zarówno z Df0: (do dysków amigowskich), jak i z Df10: (do dysków PC) zamienne — w zależności od tego, w jakim formacie jest dyskietka, jednak we wczesnych wersjach programu nie jest to takie oczywiste. Wersja 5.0 CrossDOS-a pozwala natomiast na wykorzystywanie dysków HD (1,44MB), ale jedynie na komputerach wyposażonych firmowo w taką stację (chodzi dokładnie o nowsze egzemplarze A3000) lub z dokupioną "gęstą" stacją.

KONWERTERY GRAFICZNE

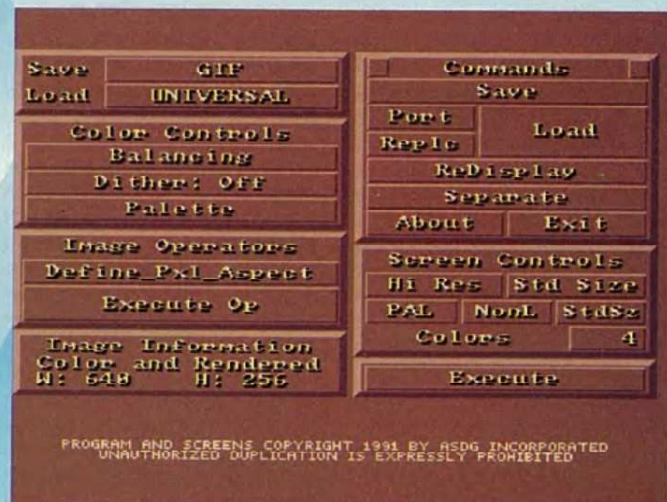
Wiemy już, w jaki sposób przetrzucać pliki. Jednak, czy PC odczyta grafikę zrobioną na Amidze? Niektóre programy na pewno tak, ale

zwykle programy mają pewne ograniczenia, na przykład niemożność odczytania obrazków w trybie HAM czy w tzw. overscanie. Pewną pomocą może być tu program o nazwie Link Image lub Art Department Pro.

Oba programy potrafią zamienić zwykły obrazek IFF IIBM na wiele formatów szeroko stosowanych w "niebieskim" świecie — między innymi GIF, PCX, TIFF czy BMP. Przy okazji jednak trzeba dodać, że

li zamierzamy dalszą obróbkę prowadzić za pomocą Quattro. W tym ostatnim przypadku należy uruchomić Quattro rozkazem "q123");

* Jeśli mamy bazę danych przygotowaną programem MI-AmigaFile, wówczas najpierw należy ją na Amidze "wydrukować na zbiór" rozkazem PRINT TO FILE, a następnie tak uzyskany zbiór ASCII przetrzucić za pomocą DOS-2-DOS na IBM-a. Pewien problem może wystąpić wówczas, gdy baza danych w



programy te nie przenoszą plików na dysk PC — operację tę trzeba wykonać programem na przykład DOS-2-DOS.

INNE "EMULACJE MINIMALNE"

Niektóre programy Amigowskie nie są emulatorami, nie mają też możliwości kopiowania ze zmianą formatu, jednak też mogą posłużyć nam jako pomoc naukowa do lekcji "IBM na Amidze".

Należy tu między innymi program o zagadkowej nazwie VIP Professional. Nie jest to nic innego jak zaimplementowana na nasz wspaniały komputer najstarsza wersja programu Lotus 1-2-3. Działa dokładnie tak samo. Co należy zrobić, jeśli chcemy dalej pracować na IBM-ie z danymi przygotowanymi na Amidze? Wystarczy:

* przegrać zbiory z VIP-a o rozszerzeniu ".WKS" za pomocą programu DOS-2-DOS; * zmienić końcówkę zbioru na ".WK1" (jeśli chcemy pracować na Lotusie) lub "WQ1" (jeże-

MiaFile zawiera polskie znaki. Amigowski standard (ks. Pikula lub prof. Bruszewskiego) zawiera znaki o kodach wyższych niż 128, które DOS-2-DOS przeniesie bez problemów. Niestety, kody polskich liter na Amidze są inne niż kody na IBM. Sposób, w jaki można sobie z tym poradzić, opisany jest niżej;

* Teksty napisane w popularnym Cygnusie również mogą stworzyć pewien problem w zakresie polskich znaków. Nie ma jednak sytuacji bez wyjścia. Przypuścimy, że nasz wydawca zażyczył sobie, aby dostarczony mu tekst był napisany w nieszczęsnym "standardzie" Mazovia. Co robić? Nie przejmując się tym, że zapisany za pomocą Cygnusa zbiór ASCII zawiera kody wyższe od 128 — mamy w zależności od naszych "źródeł" dwa wyjścia:

— skopiować zbiór programem Tekston autorstwa Tomasza Kokoszczyńskiego (być może pojawiła się już jego w pełni dopracowana wersja). Przerzucić nam dane od razu na właściwy

format. Sęk w tym, że współpracuje jedynie z Amigowskim standardem ks. Pikula. Jeżeli nasz tekst jest napisany w innym standardzie, należy wcześniej dopasować odpowiednio zbiór Amigowski i uśmiechnąć się do prof. Bruszewskiego, aby odstąpił program o nazwie PL>PL konwertujący standardy w ramach Amigi; — jeżeli nie mamy dostępu do w/w opisanych programów pomocniczych, możemy wybrać nieco bardziej skomplikowane wyjście:

* zaczynamy jak zwykle od przrzućenia zbioru programem DOS-2-DOS

* tak uzyskany zbiór "traktujemy" programem IBM-owskim o nazwie AmitoChi autorstwa Krzysztofa Gajewskiego i Bogusława Radziszewskiego. Mając zbiór w Chiwriterze, korzystamy z jednego z kilkunastu znanych programów przekształcających format Chiwritera na Mazovię (Chimaz autorstwa Jacka Adrjanowskiego, Chinamaz albo w ostateczności instalujemy do Chiwritera konfigurację "config.maz"- niestety nie wiem, kto jest jej twórcą).

Przejdźmy teraz do emulacji częściowej. Niestety wymaga ona pewnych inwestycji. Nakład finansowy jest znacznie większy, niż na opisywane wcześniej programy. Jaką z kart emulujących wybrać, aby z jednej strony nie zapłacić zbyt dużo, zaś z drugiej nie kupić czegoś, co nie zaspokoi naszych marzeń o udawanym IBM. Omówimy wady i zalety wszystkich spotykanych na rynku kart częściowo emulujących IBM-a. Przez emulację częściową rozumiemy karty, które emulują IBM-a nie w pełnym zakresie jego działania. Mówiąc prościej — do tego worka wrzuciliśmy wszystkie emulatory IBM-a z procesorem taktowanym wolniej niż 8 MHz dające użytkownikowi dostęp do mniejszej ilości pamięci niż "standardowe" — obecnie 640 KB.

SIDECAR

Z historycznego punktu widzenia jest to najstarsza próba pełnej (wówczas) emulacji IBM-a na Amidze. Niestety "blue brother" robi nieco większe kroki niż Amigowskie emulatory (zapewne stać go na siedmiomilowe buty), ale w

czasach - to było coś. "Przyczepka" (ang. sidecar) nazwana została tak zapewne ze względu na kształt i sposób montowania do jedynej wówczas Amigi 1000. W pudełku mieściła się karta emulująca IBM XT z procesorem Intel o taktowaniu 4,77 MHz oraz stacja dyskietek 5,25 cala, które można było sformatować na całe 360 KB. Dostarczone do niego oprogramowanie: Commodorowski MS DOS 2.11 i programy użytkowe firmy Janus pozwalały na pracę w trybie graficznym CGA, a także na kilka innych ciekawych rzeczy, których nie umiał "normalny" IBM.

A2088 i A2286

Są to następne produkty firmy Commodore, przeznaczone wyłącznie dla Amigi 2000. Pierwszy był kartą z procesorem 8088 (II) taktowanym zegarem 4,77 MHz — czyli typowy PC/XT bez turbo, nawet jeszcze gorzej (procesor). A2088 jest już produktem przestarzałym, podobnie jak klasa komputera, którego emulował, toteż zaprzestano jego produkcji. Następczynią karty A2088 została A2286 — już z procesorem 286 i taktowaniem 8 MHz. Jednak oferta firmy Commodore ma dwie poważne wady: nie jest atrakcyjna (AT/8 MHz to był przebieg kilka lat temu) i nie wytrzymuje konkurencji z produktami niezależnych firm oraz żadna z tych kart nie działa na Amidze 500 bez przeróbki tej ostatniej. Tę lukę zajęły dwa produkty — najpierw pierwszy...

KCS POWERBOARD

Jest to emulator przeznaczony wyłącznie dla Amigi 500 (ostatnio opracowano jego wersje dla A500 Plus). Wygląda jak duże rozszerzenie pamięci i jest wkładany we wnękę dolną Amigi. Urządzenie emuluje PC/XT (niestety) z procesorem V20 (odpowiednik 8086) taktowanym zegarem 8 MHz. Karta ma specjalizowany układ zastrzeżony przez firmę KCS, który odpowiedzialny jest za pracę urządzenia. Jego elektronika pozwala na uzyskanie emulacji kart

Hercules oraz CGA w czterech lub ośmiu kolorach. Ponadto KCS obsługuje dysk twardy, a na dodatek są to właściwie dwa produkty w jednym — w trybie Amigi urządzenie zachowuje się jak pospolite rozszerzenie pamięci do 1 MB... Jeżeli chodzi o pamięć, to w trybie PC producent oznajmia, że można uzyskać 704 KB pamięci i dodatkowe 64 KB jako EMS Przy okazji w zestawie znaleźliśmy od razu MS-DOS 4.01 warty, jak to dumnie jest napisane na ogłoszeniu, 130 funtów wraz z literaturą. Montaż KCS nie wymaga prawie żadnego doświadczenia elektronicznego — wystarczy włożyć kartę w odpowiednio miejsce Amigi 500.

Nowa wersja KCS Powerboard jest taktowana zegarem 12 MHz i potrafi zaemulować kolorowo kartę VGA.

ATONCE

Jest to druga karta proponowana posiadaczom Amigi, którzy chcą być "kompatybilni". W tym przypadku jednak jest to produkt, który można wykorzystać zarówno w Amidze 500, jak i w Amidze 2000. Emuluje on PC/AT z procesorem 80286 taktowanym zegarem... 7,14 MHz! Dzieje się to dlatego, że karta nie ma własnego kwarca taktującego i w całości współpracuje z Amigą, która, jak wiadomo, jest taktowana podaną częstotliwością. Tu dociekliwi Czytelnicy mogą zapytać, czy w takim razie, jeżeli mamy tzw. dopalacz, czyli układ(y) przyspieszający pracę komputera, to działać on będzie z ATOnce? Otóż nie — a to "dzięki" temu, że ATOnce ma własny procesor MC68000, na którym (!) zamontowana jest płytką z procesorem 80286 i układem specjalizowanym. Oznacza to niemożność wykorzystania dopalaczy, nawet w trybie Amigi! Jest to jedna z największych wad ATOnce — poza tym powstaje dylemat: "co zrobić z procesorem, który jest w komputerze?" Dowcipniejsi radzą oprawić go w ramki i szkło i powiesić na ścianie...

Powracając jednak do prezentacji możliwości ATOnce, można wymienić jego zalety graficzne: emulacja kart Her-

cules (w trybie interlace), CGA, VGA (ale w trybie mono) oraz Olivetti i Toshiba 3100 (obie również w interlace). Urządzenie wykorzystuje złącza szeregowo i równoległe Amigi jako odpowiadające złączom w PC. Ponadto mysz Amigi może być traktowana przez ATOnce jako Microsoft mouse zainstalowana na COM1 lub COM2. Wszystko to można ustawić w specjalnym programie konfiguracyjnym dołączonym do zestawu. Producent nie dodał jednak żadnej wersji MS-DOS, co zmusza przyzwoitego użytkownika do nabycia systemu operacyjnego osobno. Według mnie jest to poważny błąd handlowy. Przeciwstawić jemu można jednak zaletę, która nie została dotąd wymieniona, mianowicie ATOnce może pracować równoległe z Amigą! Jest to największa zaleta tego emulatora. (Jak widać, panuje moda na "2 in 1"). Z ciekawostek można dodać, że możliwe jest wykorzystanie pamięci CMOS — tak jak w prawdziwym AT. Montaż ATOnce jest troszkę bardziej skomplikowany niż KCS Powerboard — użytkownik musi otworzyć Amigę, co jak wiadomo równoważne jest z utratą gwarancji (włożenie KCS, na mocy umowy z Commodore, tego nie powoduje, choć w naszych warunkach nie jest to oczywiste — wszak w serwisie mogą powiedzieć inaczej). Wymiana procesora nie jest trudna (jeżeli nie robi się tego za pomocą młotka i tómu), ale jeżeli przypadkiem mamy twardy dysk A590 w starszej wersji, zmuszeni będziemy do wyjęcia układu Gary i podłożenia pod niego specjalnej podkładki umożliwiającej poprawną pracę systemu. ATOnce może wykorzystywać prawie całą pamięć Amigi — proporcje określa się w programie konfiguracyjnym. Ostatnio pojawiły się dwie nowe wersje ATOnce — jedna dla Amigi 500 Plus oraz druga dla A2000/A3000, posiadająca procesor 80386SX i pracująca z prędkością 25 MHz. To już jest ciekawa alternatywa dla chcących mieć i rybki, i akwarium — dobry komputer i (brzydkie słowo) kompatybilność. edn.

Wśród użytkowników Amigi w Polsce jest dość rozpowszechniony standard polskich znaków nazywany zwykle od nazwiska autora "standardem ks. Pikula". Jak doszło do powstania tego standardu? Co zdecydowało o takim, a nie innym, wyborze kodów dla polskich znaków? Dlaczego jest kilka różnych standardów, gdy ideałem byłoby istnienie tylko jednego? Na te pytania postaram się krótko odpowiedzieć.

Jan Pikul

Gdy w sierpniu 1988 w czasie pobytu w RFN kupiłem Amigę z drukarką Star NL- 10, postanowiłem od razu, że mój komputer będzie pisał po polsku. Już wtedy, na samym początku, poza granicami kraju, stanąłem wobec problemu wyboru kodów dla polskich znaków. Standard to właściwie kody. Do kodów trzeba dopasować mapy klawiatury, fonty, sterowniki drukarek. Wybierając kody dla polskich znaków, kierowałem się następującymi zasadami:

- powinny to być kody z zakresu 160-255, by nie ingerować w podstawowy zestaw ASCII i ominąć kody sterujące 128-159, których w Amidze nie da się wykorzystać bez głębokich zmian w systemie (z tego powodu nie można przyjąć żadnego ze standardów PC, łącznie z lansowanym ostatnio Latin2);

- znaki powinny mieć kolejność alfabetyczną;

- kod małej litery powinien być o liczbę 32 większy od kodu odpowiedniej dużej litery (łatwiej jest wtedy pisać programy);

- polskie znaki nie mogą kolidować ze znakami niemieckimi.

Uważam, że te zasady są aktualne i dziś. Wtedy, gdy nie potrafiłem jeszcze zmienić fontów systemowych i musiałem się liczyć z nieobecnością polskich znaków w wielu miejscach ekranu, przyjąłem jeszcze jedną zasadę, którą dziś można by uznać za nieaktualną:

- polskie znaki powinny być zastąpione, jeżeli to możliwe, przez znaki do nich podobne.

Ta ostatnia zasada sprawiła, że "ć" weszło w miejsce "cedilla", co jest przez niektórych uważane za koronną wadę przyjętego przeze mnie standardu.

Ostatecznie dobór kodów wygląda następująco:

Ą - 198 - \$C6	ą - 230 - \$E6
Ć - 199 - \$C7	ć - 231 - \$E7
Ę - 202 - \$CA	ę - 234 - \$EA
Ł - 206 - \$CE	ł - 238 - \$EE
Ń - 209 - \$D1	ń - 241 - \$F1
Ó - 211 - \$D3	ó - 243 - \$F3
Ś - 213 - \$D5	ś - 245 - \$F5
Ż - 219 - \$DB	ż - 251 - \$FB
Ż - 222 - \$DE	ż - 254 - \$FE

Pracę nad spolszczeniem systemu Amigi podjąłem początkowo dla własnych potrzeb, nie mając kontaktów z innymi

AMIGA A SPRAWA POLSKA - STANDARD XJP

fanami tego komputera, nie wiedząc również nic o podobnych wysiłkach innych. Od początku przyjąłem zasadę: jak najmniej zbędnej ingerencji w system operacyjny komputera, ale jednocześnie konieczne zmiany przeprowadzić na tyle głęboko, by polski system był "przyjazny dla użytkownika" i działał ze wszystkimi programami. W miarę zdobywania doświadczenia tworzyłem coraz nowsze elementy systemu, ale zasadnicza koncepcja pozostała bez zmian. Gdy dowiedziałem się o istnieniu innych standardów, mój system był już na tyle rozpowszechniony, że po konsultacji z doświadczonymi amigowcami postanowiłem go nie zmieniać.

Jak dzisiaj wygląda opisywane środowisko operacyjne?

Zacznę od wejścia, czyli od klawiatury. Obecność polskich znaków na klawiaturze zapewnia polska mapa klawiatury, podstawiana (zwykle już w startup-sequence) systemowym poleceniem SetMap. Mapa klawiatury może być w zasadzie dowolna, to zależy od upodobań użytkownika, ważne jest tylko, by zawierała odpowiednie kody dla polskich znaków. Dla mojego standardu opracowałem dotychczas trzy klawiatury:

1. "d-pol" to klawiatura niemiecko-polska, opracowana najwcześniej. Polskie znaki osiąga się przez alt+klawisz, duże przez alt+shift+klawisz. Ze względu na kolejność alfabetyczną i analogię znak "z" umieściłem na klawiszu "z", natomiast "ż", dla którego trzeba było znaleźć inne miejsce — na klawiszu "v"



(backslash) — tak jak w polskiej maszynie do pisania. Zupełnie przypadkowo "z" znalazło się też pod alt+t.

2. "usa-pol" to klawiatura USA-polska, opracowana, gdy mój standard zaczął się upowszechniać i zaszła potrzeba dostosowania go do komputerów z angielską klawiaturą. Układ znaków jest identyczny, jak na rozpowszechnionej klawiaturze PC: polskie znaki i tu osiąga się przez alt+klawisz, jednak "z" jest pod "x", a "ż" pod "z".

3. "d-pol1" to klawiatura opracowana ostatnio. Jej potrzebę zasugerował mi prof. Bruszewski. Różni się od "d-pol" innym układem "z" i "ż". Mianowicie "z" jest pod alt+x, a "ż" pod alt+z, jak na klawiaturze "usa-pol" (tyle, że klawisz "z" jest tu w innym miejscu, zgodnie z normą DIN).

Najczęściej stosowanym urządzeniem wyjściowym jest ekran komputera. Tu obecność polskich znaków zapewnia polecenie PolFonts, które zamienia fonty systemowe "topaz 8" i "topaz 9e" na ich polskie odpowiedniki o nazwach "tpzpl 8" i "tpzpl 9e". Te nazwy to relikty z czasów, gdy PolFonts jeszcze nie istniało, a taka właśnie długość nazwy ułatwiała przeróbki programów (obecnie zupełnie zbędne). Polecenie PolFonts jest krótsze i działa w tym wypadku lepiej niż "importowane" polecenia ChgFont i SetFont. Obecna wersja pracuje także z systemem 2.0. W wypadku użycia FF (=FastFonts) PolFonts musi być wywołane wcześniej. Zresztą nowsze wersje polecenia FF mogą podstawić dowolne fonty nieproporcjonalne i skutecznie zastępują PolFonts zwłaszcza wtedy, gdy komuś zależy na podstawieniu jakiegoś "egzotycznego" fontu. Pol-Fonts bowiem, w myśl zasady "jak najmniej ingerencji", może podstawić tylko "swoje" fonty. Wiele programów korzysta z fontów dyskowych. Te fonty też powinny zawierać polskie znaki pod odpowiednimi kodami. Wszystkie fonty dostępne standardowo na dyskietce "Workbench" mają już swoje polskie odpowiedniki. Poza tym istnieje sporo innych fontów spolszczonych w opisywanym standardzie. Ciągłe jednak istnieje potrzeba opracowywania nowych polskich fontów, w tym także fontów kolorowych i wektorowych.

W praktyce najważniejszym, ale i najtrudniejszym zadaniem okazało się napisanie polskich sterowników dla drukarek. Właśnie pomyślnie rozwiązanie tego problemu zapewniło opisywanemu standardowi popularność i rozpowszechnienie. Sprawa sterowników na pewno nie jest jeszcze zamknięta, ciągle rodzą się nowe potrzeby, a ich rozwiązywanie jest bardzo czasochłonne, ale już w tej chwili istnieją działające sterowniki do większości popularnych drukarek. Nie licząc kilku prototypowych specjalnych driverów, do aktualnego zestawu należą:

A. Sterowniki tłumaczące kody polskich znaków na inny standard (wymagają zainstalowania polskich znaków w drukarce):

1. "EpsonX_Mzv" — dla drukarek 9-igłowych standardu Epson, z polskimi znakami w standardzie Mazovii. Sterownik ma poprawiony błąd swego poprzednika "Star_Mzv", który odmawiał posłuszeństwa po druku graficznym lub barwnym.
2. "EpsonQ_Mzv" — jak wyżej, jednak dla drukarek 24-igłowych.
3. "IBM_Mzv" — dla drukarek standardu IBM Proprinter z zestawem Mazovii.

4. "D-100M_6" — dla polskiej drukarki D-100M z zestawem znaków "Latin2".

B. Sterowniki wykorzystujące znaki definiowalne przez użytkownika w draft i NLQ:

5. "StarNLpol" — dla drukarek Star NL-10, NX-15 i podobnych.
6. "StarLCpol" — dla drukarek Star LC-10, LC-20, NX-1000 i podobnych.
7. "StarNEWpl" — dla drukarki Star LC-200 i podobnych.
8. "EpsonLXpl" — dla drukarek Epson LX-400, LX-800 i podobnych.
9. "NL-10PL" — dla drukarki Star NL-10 (Centronics) — szybki druk.
10. "SeikoshaSP_PL" — dla drukarek Seikosha SP-1000 i SP-2000.

C. Sterowniki definiujące znaki draft i emulujące NLQ za pomocą grafiki (dla drukarek, które pozwalają definiować tylko znaki draft):

11. "EpsonXgrafPL1" — kursywa draft zdefiniowana w sterowniku.
12. "EpsonXgrafPL2" — kursywę draft tworzy drukarka. Sterowniki tej grupy mogą wymagać indywidualnego dopasowania do drukarki.

D. Sterowniki typu "Generic" — bez grafiki, bez możliwości programowego sterowania drukarką.

13. "GenericPol_over" — dla wszystkich typów drukarek, polskie znaki tworzy przez nałożenie dwóch znaków ASCII.
14. "GenericPol_def" — definiuje polskie znaki (tryb draft).

Bezwzględnie najlepszym rozwiązaniem jest wyposażenie drukarki w generator znaków z polskimi znakami narodowymi. Standard tych znaków jest w zasadzie obojętny — sterownik może "przetłumaczyć" jeden standard na drugi. W tej chwili istnieją sterowniki dla drukarek z zestawem znaków Mazovii, jednak w każdej chwili mogą być dostosowane do innego standardu.

Jednak i użytkownicy drukarek, w których nie zainstalowano polskich znaków, mogą w większości przypadków sprawnie drukować po polsku. W najlepszej sytuacji są w tej chwili posiadacze popularnych drukarek firmy Star, którzy niemal nie odczuwają braku polskiego EPROM-u: ich drukarki drukują polskie litery, bez jawnej inicjacji i bez większego opóźnienia, dzięki odpowiednim sterownikom.

Ocenę mojego rozwiązania pozostawiam Czytelnikom, których zapewne wielu korzysta z opisanych elementów systemu operacyjnego. Zwykle wystarczy wystartować ze spolszczonej dyskietki "Workbench", by nowy program zaczął od razu działać i drukować po polsku. Właśnie to było kiedyś moim marzeniem, gdy rozpoczynałem pracę nad polskim środowiskiem operacyjnym dla Amigi. Istnieją na pewno lepiej przemyślane (gdy chodzi o dobór kodów polskich znaków) propozycje standardu, na przykład propozycja prof. Bruszewskiego, która oszczędza "cedilla". Jednak atutem mojego rozwiązania są działające sterowniki drukarek — i związane z tym rozpowszechnienie. Czy w tej sytuacji zmieniać standard, czy pozostać przy istniejącym? Który ze standardów powinien pozostać jako jedyny? I na to pytanie niech pomogą odpowiedzieć Czytelnicy.

KLAWIATURA Z ALASKI

Utworzenie polskiego standardu na Amigę wiąże się z podjęciem decyzji o numerach kodów osiemnastu znaków: ąćęłńóśźż, ȺĆĘŁŃÓŚŹŻ. Standard niesie ze sobą bardzo określone korzyści (patrz Magazyn AMIGA nr 1/92), ale jednocześnie standard to reżim; to zaprzeczenie idei wolności, rozumianej w specyficznie polski sposób. Standard to narzędzie komunikacji. Z tej definicji wynika, że powinien on być jeden. Natomiast klawiaturą jako narzędzie standardu, może reprezentować najdziksze pomysły.

Wojciech Bruszewski

Fizycznie rzecz biorąc, klawiatura na Amidze miała dotychczas trzy, zasadniczo różne, wcielenia:

- premierowe, bardzo zgrabne w Amidze 1000;
- zbliżone do standardu klawiatury na PC w A-500, A-2000 i następujących modelach;
- z obciążonym blokiem numerycznym w Amidze 600.

Dzisiejsze moje sympatie idą w kierunku klawiatury A-600, ale rozsądek podpowiada, aby trzymać się standardu A-2000 ze względu na istnienie niewielkiej grupy programów, które w specyficzny sposób "widzą" klawiaturę, w tym blok numeryczny.

2. Na fizycznie ukształtowanej klawiaturze namalować można różne znaki i w ten sposób powołać do życia klawiaturę narodową.

Jeżeli pierwsze litery na klawiaturze dają się złożyć w słowo

QWERTY — to jest to odmiana amerykańska (ale niekoniecznie),

QWERTZ — niemiecka,

AZERTY — francuska.

Z uwagi na polskie przyzwyczajenia, przeniesione z mechanicznej maszyny do pisania, najbliższa nam powinna być klawiatura niemiecka.

3. Fizycznej klawiaturze należy programowo przypisać numery kodów JAKIEGOŚ *) standardu.

Czynność zainstalowania mapy klawiatury odbywa się automatycznie podczas startu maszyny.

Rozkaz w startup-sequence wygląda tak:

Sys:System/SetMap d

co oznacza, że wpisaniem nowej mapy klawiatury zajmuje się program SetMap przechowywany w szufladzie System Workbench, a wpisywany jest niemiecki wzorzec pod nazwą d przechowywany w szufladzie DEVS/KEY-MAPS.

Polski nabywca Amigi, po otwarciu kartonowego pudła, może zobaczyć jedną z możliwych klawiatur narodowych, a na załączonych dyskietkach może dogrzebać się do wszystkich standardowych KEYMAPS.

Prawdopodobieństwo zobaczenia klawiatury polskiej oraz polskiej keymap jest bliskie zera.

4. Tim Friest jest mieszkańcem Anchorage na Alasce. Tim nie wie, że jego praca może rozwiązać poważną część narodowego problemu w bardzo odległym od Alaski kraju. Dowie się tego z miesięcznika AMIGA, jeżeli znajdzie w Anchorage kogoś, kto zna język polski.

Tim jest autorem programu KeyMapEd.

KeyMapEd umożliwia realizację różnych keymaps.

W dokumentacji do KeyMapEd, Tim informuje nas, że na Amidze w zasadzie każdy klawisz może przechowywać kilka macro-instrukcji w zależności od tego, w jaki sposób jest naciskany, czy:

- jako ALONE, czyli sam
- z klawiszem [SHIFT]
- z klawiszem [ALT]
- z klawiszem [CTRL]
- z klawiszami [SHIFT] + [ALT]
- z klawiszami [CTRL] + [SHIFT]
- z klawiszami [CTRL] + [ALT]
- z klawiszami [CTRL] + [SHIFT] + [ALT]

ponadto każdy klawisz może być:

CAPSABLE — klawisz po wciśnięciu [CAPSLOCK], zachowuje się tak, jakby klawisz [SHIFT] był permanentnie wciśnięty;

REPEATABLE — klawisz powtarza znak tak długo, jak długo jest wciśnięty;

DEAD/DEADABLE — kombinacja klawiszy, która wywołuje na ekran znaki narodowe z akcentami.

Rozróżnia się cztery rodzaje klawiszy:

NOP klawisz "bez wartości" — nic nie robi;

NORMAL — klawisz, który przechowuje do czterech macro-instrukcji definiujących pojedynczy znak;

STRING — klawisz, który przechowuje do ośmiu macro-instrukcji o długości do 32 znaków (dotyczy Amiga DOS v.1.3);

DEAD — który zachowuje się na trzy sposoby:

- jest DEAD, czyli "martwy", i modyfikuje jedynie wartość klawisza, który zostanie po nim naciśnięty;
- jest DEADABLE, czyli podlega wpływom klawisza DEAD;
- jest odporny na wpływ klawisza DEAD.



Wbrew pozorom klawisz typu DEAD nie tworzy znaków przez "nadbitkę", ale pozwala wywołać na ekran akcentowany znak, którego w danym momencie nie ma na klawiaturze. Klawisze takie są częścią amerykańskiej mapy USA1, a nie ma ich na niemieckiej mapie d, gdzie są akcentowane znaki, na przykład (ä i ü). Jestem zdania, że na polskiej klawiaturze stosowanie klawiszów DEAD nie ma sensu.

KeyMapEd v.1.02 jest programem Public Domain. Znaleźć go można na dyskietce Freda Fisha nr 193.

5. Na polskim komputerze używa się alternatywnie dwóch map klawiatur, ale na Amidze może być ich więcej.

Na klawiaturze MASZYNISTKI polskie znaki umieszczone są zgodnie z układem znaków na mechanicznej maszynie do pisania. Przesiadka z maszyny do pisania na komputer jest dla maszynistek stosunkowo bezbolesna. Polskie ączęćróśź są widoczne na klawiszach.

Na klawiaturze PROGRAMISTY polskie znaki wywołuje się przez wciśnięcie klawisza [ALT], według zasady:

[ALT] + [A] = [ą]
 [ALT] + [SHIFT] + [A] = [Ą]

Problem dwóch znaków z literą "z", czyli "ź" i "ż" przyjęto się rozwiązywać przez przypisanie "ź" literze "x",

[ALT] + [X] = [ź]
 [ALT] + [SHIFT] + [X] = [Ż]

Mapa klawiatury PROGRAMISTY, zainstalowana na niemieckiej klawiaturze, pozwala "z biegu" pisać w trzech językach: angielskim, niemieckim i polskim.

Trzeci rodzaj to klawiatury DZIWNE. Publikując swego czasu moją propozycję standardu, który dziś nazywam AmigaPL, zaproponowałem powołanie do życia klawiatury MASZYNISTY. Jest ona zgodna z ideą klawiatury PROGRAMISTY, ale dodatkowo umieściłem polskie znaki na bloku numerycznym według zasady:

[1] = [ą]
 [1] + [SHIFT] = [Ą]
 [1] + [ALT] = [1]

Polskich znaków jest akurat 9 + 9. Pomysł ten zrodził się w momencie, gdy do pisania używałem programu Excellencel, który nie widział klawiszy [ALT].

6. Zdefiniowanie polskiej mapy za pomocą programu KeyMapEd jest niezwykle proste. Ale zanim przystąpi się do pracy, należy:

- a) Znać adresy polskich znaków jako liczby szesnastkowe; innymi słowy zdecydować się na jakiś standard;
- b) Podjąć decyzję, jakiego typu klawiaturę chcemy zrealizować.

Urchamiając KeyMapEd, należy wpisać z dysku istniejący wzorzec, któremu fizycznie odpowiada klawiatura, jaką mamy ochotę zmodyfikować. Ekran KeyMapEd jest obrazem klawiatury typu A-500/A-2000 **). Adres polskiego znaku "ą" na klawiaturze przypisuje się według zasady:

\xe2

gdzie \x jest macro-instrukcją sygnalizującą pojedynczy znak, a e2 jest liczbą szesnastkową odpowiadającą adresowi "ą" w standardzie AmigaPL.

Na klawiaturach DZIWNYCH możliwe jest umieszczanie pod jednym klawiszem dłuższych sekwencji znaków, na przykład rozkazów Amiga DOS.

Można sobie wyobrazić, że klawisz [HELP] (z altem) klawiatury o nazwie Alaska ma przypisany rozkaz:

sys:System/SetMap\x20Polska\r

a ten sam klawisz [HELP] (z altem) klawiatury Polska ma przypisany rozkaz:

sys:System/SetMap\x20Alaska\r

W otwartym oknie SHELL lub CLI uderzenie w klawisz [HELP] przy wciśniętym [ALT] będzie szybkim przeskokiem od jednej mapy do drugiej. W programie KeyMapEd macro-instrukcja \r symuluje uderzenie w klawisz [RETURN], a \x20 jest spacją.

Zmodyfikowaną mapę zapisujemy na dysku z nową nazwą.

W wypadku keymaps nazwa zostaje zapisana również wewnątrz pliku. Nie zaleca się więc jej późniejszych zmian, ponieważ wewnątrz pliku pozostanie stara nazwa. Amiga DOS mapę klawiatury ładuje do pamięci tylko raz i rozpoznaje według wewnętrznie zapisanej nazwy. Po zmianie mapy, w pamięci przechowywane są obie, poprzednia i aktualna. Wcześniej załadowaną mapę ponownie wywołuje się z pamięci, a nie z dysku. Trzeba o tym pamiętać podczas pracy nad nową mapą klawiatury. Wpisana do pamięci raz, nie będzie się zmieniać pomimo zmian dokonywanych w KeyMapEd. Nową mapę pod tą samą nazwą można przetestować po ponownym starcie systemu.



Ekran programu KeyMapEd z lewej strony okno klawisza [a] z przypisanymi adresami dla [ą] i [Ą]. Z prawej — okno klawisza [HELP] z dodatkową instrukcją. Polskie znaki pojawiają się automatycznie, ponieważ systemowy Topaz wymieniony został na spolszczoną czcionkę TopPL.

*) Muszę napisać JAKIEGOŚ standardu, ponieważ DEMOKRATYCZNIE nie mogę napisać JAKIEGO. Wolni ludzie z USA, Francji, Niemiec, Hiszpanii, ... nie mają tego problemu, ponieważ reżim nie daje im wyboru.

** W podobnie działającym programie SetKey v.1.0 widać na ekranie obraz klawiatury A-1000, co uniemożliwia zdefiniowanie niektórych klawiszy klawiatury A-2000. Taką samą wadę ma KeyToy — pokazuje aktualne "obłożenie" klawiatury typu A-1000.

ARexx, królewski język programowania

SZLACHECTWO ZOBOWIĄZUJE (cz. 2)

Markus Stoll

Ważne słowa kluczowe

"Exit", po polsku "wyjście": za pomocą tego słowa kluczowego możecie przerwać Wasz program w dowolnym miejscu. Jest to potrzebne na przykład wtedy, gdy na końcu Waszego programu głównego znajdują się jeszcze podprogramy, które po zakończeniu programu głównego nie powinny być wykonywane. Również wtedy, gdy zdarzy się błąd i dalsze wykonywanie programu byłoby bezsensowne; także w tym wypadku możecie przerwać program za pomocą "exit" i umieścić po słowie kluczowym dodatkowo numer błędu, który będzie przekazany do Amiga DOS-u.

Pętle wszelkiego rodzaju

Pętle są w językach programowania ważnymi elementami strukturalnymi, dzięki którym można określić części programu powtarzać w sposób kontrolowany. Bloki programowe, do których zaliczają się także pętle, są zasadniczo zamknięte między słowami kluczowymi "do" i "end". Jeśli po "do" następuje jeden z możliwych parametrów pętli, wewnętrzny blok programu będzie wykonywany, aż wypelni się warunek zakończenia pętli. Listing 10. (na końcu artykułu) ukazuje

Pierwszy odcinek kursu zaznajomił Was z podstawami ARexxa. Wiecie, jak się tworzy i uruchamia programy ARexxa. Znać składnię języka, umieć posługiwać się zmiennymi i operatorami. W tym odcinku kursu poznamy najważniejsze elementy zbioru rozkazów ARexxa. Przy okazji zobaczycie, że ARexx nie pozostaje w tyle za innymi językami wysoko ocenianymi pod względem możliwości programowania strukturalnego. Także ten odcinek kursu jest przepelniony programami demonstracyjnymi, które będą Wam wyjaśniać wprowadzane elementy języka. Jeśli jesteście spragnieni przykładu programu użytkowego, tym razem będziecie usatysfakcjonowani: pod koniec tego odcinka poznacie na przykładzie, jak w ARexxie zaprogramować nowy rozkaz CLI o dużych możliwościach. Najpierw jednak nawiążmy do tego miejsca, w którym przerwaliśmy poprzedni odcinek kursu.

przykład dla każdego rodzaju pętli.

Chociaż komentarze w programie przykładowym wyjaśniają wszystko, chcemy niektórym punktom przyrzeć się lepiej.

Pętla licznikowa (w oryginale: Pętla FOR, czyli licznikowa; jednak takie określenie jest mylące: zob. następny typ! — przyp. tłum.)

Ten typ pętli znacie z innych języków wysokiego poziomu (jak BASIC lub C) prawdopodobnie jako pętlę FOR. Jej składnia w ARexxie jest następująca:

```
do [zmienna=wartość_początkowa]
```

```
to [wartość_końcowa] by [krok]
[... ] end
```

Przed pierwszym przebiegiem pętli przypisujecie podanej zmiennej wartość początkową. Po każdym przebiegu wartość zmiennej zostaje zwiększona o podaną wartość kroku (jeśli opuścicie "by [krok]", zostaje zwiększona o 1) i gdy wartość zmiennej przekroczy podaną wartość końcową, pętla zostaje zakończona. Za pomocą tego typu pętli możecie też zrealizować odliczanie "do tyłu". W tym celu wartość końcowa musi być mniejsza od początkowej i musi być podana

ujemna wartość kroku. Jeśli żadna wartość końcowa nie jest podana, Amiga wykonuje pętlę bez końca.

Pętla typu ARexx- FOR

Słowo kluczowe "for" deklaruje w pętlach ARexxa, ile razy, najwyżej, pętla zostanie wykonana. Parametr po tym słowie kluczowym musi przedstawiać liczbę całkowitą dodatnią. Pętla bez końca ma stałą składnię w postaci "do forever [...] end". Jak jednak dowiedzieliście się przed chwilą, jest jeszcze inna forma pętli bez końca, przy której otrzymujecie jako wartość zmiennej liczbę przebiegów pętli.

Przejdźmy do dwóch ostatnich rodzajów pętli. Przy pętlach "while" i "until" musimy nieco lepiej zaobserwować, co następuje po danym słowie kluczowym. W komentarzach jest to krótko określone jako "warunek". ARexx oczekuje tu wartości boolowskiej, a więc "prawda" lub "fałsz". To znaczy, może tu stać odpowiednia liczba (1 lub 0) podana bezpośrednio (w ten sposób można i tu osiągnąć pętlę bez końca) lub zmienna, której wartość decyduje o zakończeniu albo kontynuowaniu pętli. Normalnie jednak, po słowie kluczowym nastąpi jako warunek operator porównania (pamiętamy z pierwszej części kursu, że operatory porównania zwracają wartości boolowskie). Pętla:

```
do for 5 while i=0 [...] end
```

będzie tak długo wykonywana, jak długo zmienna "i" ma

wartość 0, jednak nie więcej niż 5 razy. Teraz już na pewno rozumiecie też, dla czego przy objaśnieniu słowa "for" zostało wspomniane, że ustala ono maksymalną liczbę przebiegów pętli: w pętlach można umieszczać jednocześnie różne warunki zakończenia pętli.

Gdzie tu jest wyjście?

Przy uruchamianiu naszego programu przykładowego zwróciliście może uwagę, jak można wymusić nadprogramowe opuszczenie pętli. Pętla bez końca "do forever [...]" została zakończona po sześciu przebiegach za pomocą polecenia "leave". Gdy ARexx natrafi na słowo kluczowe "leave", aktualna pętla zostaje opuszczona bez względu na warunki. Gdy znajdujecie się wewnątrz wielu zagnieżdżonych pętli, możecie po słowie "leave" podać zmienną licznikową tej pętli, po której "end" — interpreter — ma kontynuować pracę. Istnieje także drugie polecenie o prawie takim samym działaniu. Słowo kluczowe "break" powoduje wyjście z aktualnego bloku "do [...] end". Te nowe zjawiska wyjaśnia listing 11.

Wraz z pętlami poznaliśmy już jedną formę warunkowego wykonania programu. Teraz zobaczycie, jak w ARexxie jest zaimplementowana znana instrukcja warunkowa "if": (zobacz listing 12).

Polecenie "if" oczekuje jako argumentu wartości boolowskiej, podobnie jak w wypadku pętli "until" i "while". W naszym przykładzie użyliśmy znów operatorów porównania. Pierwsza instrukcja warunkowa "if" sprawdza, czy wartość zmiennej jest większa od 4. Ponieważ to zachodzi, polecenie po słowie kluczowym "then" zostaje wykonane, a polecenie po słowie kluczowym "else" opuszczone. Drugi przykład pokazuje, co należy zrobić, gdy chcecie w jednej instrukcji warunkowej "if" umieścić cały ciąg rozkazów. Należy ciąg rozkazów przekształcić w blok przez zamknięcie go między poleceniami "do" i "end". Trzeba jeszcze wspomnieć, że słowo kluczo-

we "else" możecie opuścić, gdy go nie potrzebujecie.

Wolny wybór

Instrukcja warunkowa "if" umożliwia programiście warunkowe podzielenie programu na dokładnie dwie gałęzie (albo polecenie po "then", albo polecenie po "else"). Jak w życiu słowa "tak" i "nie" często nie wystarczają do udzielenia pełnej odpowiedzi, tak i w programowaniu często potrzebne jest rozróżnienie wielu przypadków. W tym celu możecie zagnieżdżać kilka instrukcji warunkowych "if". Bardziej pogłębiony jest jednak kolejny element języka ARexx: instrukcja "select". Przykład — listing 13.

Pomiędzy słowami "select" i "end" możecie za pomocą szeregu instrukcji "when" (każdorazowo z argumentem, zwracającym wartość boolowską), rozróżniać różnorodne przypadki. Za każdym razem zostanie wykonane najwyżej jedno odgałęzienie wewnątrz grupy "select". Na końcu może stać także słowo kluczowe "otherwise". Oznacza ono "w przeciwnym wypadku"; stojące po nim polecenie (lub grupa poleceń) zostanie wykonane tylko wtedy, gdy żaden z warunków "when" nie będzie spełniony.

Podprogramy i programy pomocnicze

Podział na jeden program główny i mniejsze podprogramy jest zasadniczym elementem modularyzacji programów. W ARexxie możecie stworzyć zarówno podprogramy, które mają dostęp do zmiennych globalnych programu głównego, jak i pisać procedury, pracujące ze zmiennymi lokalnymi. ARexx rozpoznaje podprogram po jego nazwie. Nazwy podprogramów stoją, podobnie jak rozkazy, w tekście programu, jednak są zakończone dwukropkiem ":". Program przykładowy z listingu 14. tłumaczy różnicę między podprogramem ze zmiennymi globalnymi i procedurą ze zmiennymi

lokalnymi. Przypatrzmy się krok za krokiem, co ten program robi. Najpierw przypisujemy zmiennej "test" łańcuch znaków "Halo!". Następnie za pomocą polecenia "call" wywołujemy podprogram "procedura". Interpreter poszukuje wtedy w programie podanej nazwy podprogramu, po której, jak mówiliśmy, musi stać dwukropek. Ponieważ po dwukropku w tym przypadku znajduje się słowo kluczowe "procedure", ARexx tworzy dla podprogramu nowy obszar zmiennych. Możecie to poznać po następującej teraz komendzie "say test", która nie powoduje wyprowadzenia "Halo!". Ponieważ zmienna "test" w tym podprogramie nie jest jeszcze podstawiona, otrzymacie jako wynik nazwę zmiennej "test". Teraz zmienna "test" zostaje na nowo podstawiona i opuszczamy podprogram rozkazem "return". Jednocześnie ARexx aktywuje obszar zmiennych programu głównego i dlatego rozkaz "say test" daje teraz wynik "Halo!".

Poznaliśmy właśnie sposób użycia zmiennych lokalnych w podprogramach ARexxa. Teraz następuje prawie to samo jeszcze raz. Jednak teraz zostaje wywołany podprogram "podprogram", który nie jest opatrzony słowem kluczowym "procedure". Teraz podprogram i program główny mają ten sam obszar zmiennych, tak że dwa rozkazy umieszczone w podprogramie mają taki sam skutek, jak gdyby stały w programie głównym. W tym programie przykładowym zauważacie także, do czego może służyć rozkaz "exit". Podprogramy stają się jednak dopiero wtedy naprawdę interesujące, gdy można do nich przekazywać argumenty i otrzymywać z powrotem wynik. Tego potrzebujecie na przykład, by stworzyć nową funkcję matematyczną. Pięknym przykładem, pozwalającym wytłumaczyć te możliwości, jest funkcja "silnia". Silnia liczby całkowitej (dodatkowo — przyp. tłum.) jest to iloczyn wszystkich liczb całkowitych od 1 do danej liczby. Silnia może być obliczona zarówno przy użyciu pętli, jak i rekurencyjnie (w ten sposób,

że funkcja wywołuje samą siebie z nowym argumentem). Listing 15 ukazuje obydwie możliwości. Jak widzicie, funkcje można wywołać na dwa sposoby. Gdy oczekujecie wyniku, możecie wywołać funkcję zgodnie z następującą składnią:

```
wynik = funkcja(argument)
```

Interpreter ARexxa szuka w programie odpowiedniej nazwy funkcji, wywołuje funkcję z podanym argumentem i przypisuje danej zmiennej wartość, która w podprogramie została podana po słowie kluczowym "return". Jak to już zostało opisane w programie 15., możecie podać równocześnie także więcej argumentów. Oto przykład wywołania funkcji z dwoma argumentami:

```
wynik =
funkcja(argument1, argument2)
```

Polecenie "call", które już poznaliśmy, może być także wykorzystane do wywoływania funkcji. Wartość, którą funkcja zwraca jako wynik, zostaje wtedy przypisana automatycznie zmiennej "result". Oto przykład wywołania funkcji za pomocą "call":

```
call funkcja
argument1, argument2
```

Niektóre funkcje ARexxa

ARexx dysponuje wieloma wbudowanymi już funkcjami, które pomagają Wam przy programowaniu. Dobrym przykładem funkcji wbudowanej jest:

```
addlib(nazwa, priorytet, offset, wersja)
```

Możecie wykorzystywać, obok samodzielnie napisanych funkcji, także funkcje, które inni programiści napisali już dla ARexxa. Takie funkcje są zgromadzone w bibliotekach, tzw. libraries. Za pomocą powyższego polecenia możecie nakazać interpreterowi ARexxa wczytanie wskazanej biblioteki, aby wykorzystać jej funkcje. Należy jednak zwrócić uwagę na to, że tylko takie biblioteki mogą być użyte, które są napisane specjalnie dla ARexxa. Jako argumenty należy podać nazwę biblioteki, priorytet, z jakim ARexx ma przeszukiwać daną bibliotekę, i dwa parametry charakterystyczne dla

danej biblioteki. Funkcja "addlib" zwraca wartość boolowską, która podaje, czy biblioteka była rzeczywiście dostępna. Do wyposażenia ARexxa należy biblioteka "rexsupport.library", która oferuje niektóre funkcje specyficzne dla Amigi. Dostęp do nich uzyskacie dzięki poleceniu:

```
i =
addlib("rexsupport.library", 0,
30, 0)
```

Teraz możecie wywołać na przykład (przypr. tłum.):

```
- address()
```

Ta funkcja zwraca nazwę aktualnie aktywnego portu "host". Jest to "message-port", do którego są wysyłane wszystkie te polecenia, których sam ARexx nie zna. Ten "message-port" jest Wam potrzebny do komunikacji między procesami; do tego jeszcze wrócimy.

```
- arg(liczba)
```

Nawias kwadratowy oznacza, że argument (będący liczbą) można opuścić (przypr. tłum.).

Ta funkcja bez argumentu zwraca liczbę parametrów przekazanych do aktualnie wykonywanej funkcji lub do programu głównego. Gdy podacie liczbę jako argument, otrzymacie jako wynik ten argument, którego numer odpowiada danej liczbie. Gdy na przykład Wasza funkcja została wywołana następująco:

```
i = funkcja('Test', 123)
```

możliwyście tak wykorzystać "arg()":

```
say arg() -> 2
```

```
say arg(2) -> 123
```

Następujące teraz funkcje wykonujące operacje na plikach będą nam potrzebne w programie przykładowym na końcu kursu:

```
i =
```

```
open(plik, nazwa_pliku, sposob)
```

— "open()" próbuje otworzyć podany plik. Zmienna "sposob" może mieć trzy stany: "read" dla odczytu pliku, "write" dla zapisu pliku i "append" dla dołączenia nowych danych do istniejącego pliku. "open()" zwraca wartość

boolowską, w zależności od tego, czy otwarcie pliku zostało uwieńczone sukcesem, czy nie. Możecie potem zapisywać plik lub czytać z niego, gdy w tekście programu odwołacie się do pliku za pomocą nazwy, którą podaliście jako pierwszy argument funkcji "open()".

```
— close(plik)
```

— zamyka dany plik, który wcześniej był otworzony przez "open()"

```
— eof(plik)
```

— zwraca jako wartość boolowską informację, czy został osiągnięty koniec pliku.

```
linia = readln(plik)
```

— "readln()" czyta jedną linię z podanego pliku i zwraca ją jako wynik

```
— compare(napis1, napis2)
```

— "compare()" porównuje dwa łańcuchy znakowe i w wypadku identyczności zwraca zero. W przeciwnym wypadku "compare()" zwraca numer pozycji, począwszy od której obydwa łańcuchy różnią się między sobą.

ARexx oferuje nadto o wiele więcej wbudowanych funkcji. Niektóre z nich będą nam potrzebne w przykładowym programie użytkowym na końcu tej części kursu. Wyczerpujące zestawienie wszystkich wbudowanych funkcji znajdziecie zarówno w waszym podręczniku ARexxa, jak i w "Podręczniku programowania systemowego", który jest oferowany przez Commodore z systemem operacyjnym Amiga-OS 2.0. Przeszukajcie także niektóre dyskiety "public domain". Jest wiele bibliotek ARexxa, które mogą Wam tu i tam ułatwić programowanie.

ARG i PARSE

Zauważyliście z pewnością, jak przechwyciliśmy argument funkcji w listingu 15. Moglibyśmy zamiast "arg" użyć też "i=arg(1)". Słowo kluczowe "arg" jest krótką formą od

"parse upper arg". Chcemy nieco bliżej przyrzeć się właściwościom "parse". "Parse" służy do tego, by dany łańcuch znakowy podzielić na fragmenty i przypisać te fragmenty różnym zmiennym. Instrukcja "parse" może pobrać łańcuch, który ma być podzielony, na przykład z jakiejś zmiennej (po "parse" następuje słowo kluczowe "var" i zmienna) lub z argumentu aktualnej funkcji (albo z argumentu programu głównego, który użytkownik podaje w Shellu) — za pomocą słowa kluczowego "arg". Wspomniana wyżej opcja "upper" powoduje zamianę na duże litery, zanim łańcuch zostanie podzielony (zob. listing 16).

Jeśli wywołacie ten program (listing 16 — przypr. tłum.) poleceniem:

```
rx test16 cześć
```

otrzymacie następujący wynik:

```
cześć
```

```
Chce
```

```
mi
```

```
się
```

```
pić
```

Łańcuch znaków, który przekazaliście, otrzymujecie najpierw z powrotem (rozumiecie już na pewno, jak łatwo byłoby w ARexxie napisać odpowiednik polecenia CLI "Echo"). Z kolei łańcuch znaków zostaje podzielony według zawartych w nim spacji i przypisany zmiennym "a", "b", "c" i "d". Wiodąca spacja w zmiennej "d" pochodzi stąd, że polecenie "parse" po trzeciej zmiennej po prostu przyporządkowało resztę czwartej zmiennej (razem z wiodącą spacją). Przez podanie kolejnej nazwy zmiennej moglibyście ominąć ten problem. Za pomocą "parse" możecie zlecić poszukiwanie dowolnych sekwencji znaków i według ich położenia dzielić łańcuchy znakowe. Musicie w tym celu między zmiennymi umieścić po prostu wzorzec w cudzysłowie (wzorzec odnosi

się do zmiennej, po której stoi — przypr. tłum.).

Niech i inne programy działają!

Przechodzimy teraz do najważniejszej właściwości ARexxa — do komunikacji między procesami. Za pomocą słowa kluczowego "address" i podania nazwy określicie, do jakiego message portu powinien być wysłany rozkaz. Message port to pewien adres, zgłoszony systemowi operacyjnemu wraz z nazwą przez działający program użytkowy. Możecie za pomocą polecenia "address" wysłać do takich message portów komendy, które następnie będą opracowywane przez odpowiedni program użytkowy. To jest właściwie materiał następnego odcinka kursu. Teraz chcemy się ograniczyć do pewnej formy specjalnej: "address command". Poleceniem tym możecie tak wywoływać programy, jak gdybyście się znajdowali w CLI. Listing 17 jest przykładem.

Przekazane parametry zostają rozdzielone między zmienne "zrodlo" i "przeznaczenie". Jeśli nie podano argumentu "przeznaczenie", zmienna "przeznaczenie" przyjmuje wartość dwóch znaków cudzysłowu, co sprawia, że polecenie "copy" działa zgodnie z oczekiwaniami: plik zostaje skopiowany do aktualnego katalogu.

Teraz pokażemy możliwośći ARexxa przy programowaniu pewnego rozkazu CLI. Zbudujemy program, który przeszukuje dany katalog i podaje sumę długości wszystkich plików w bajtach. Program, który to robi, wywołuje rekurencyjnie sam siebie, gdy w katalogu znajdują się jeszcze podkatalogi (listing 18). W następnym odcinku zajmiemy się sterowaniem programami wyposażonymi w port ARexxa, jak na przykład "CygnumEd".

AMIGA Magazin 8/1991

Tłum. Jan Pikuł

Listing 10

```
/* Wszystkie rodzaje pętli */
/* Najpierw blok programowy bez pętli
- dlatego "say" jest wykonane tylko raz */
do
say "Halo do!"
end
```

```
/* A teraz z parametrem pętli
1. Klasyczna pętla licznikowa */
say "Pętla licznikowa o 5 przebiegach!"
do i=1 to 10 by 2
say i
end
/* 2. Pętla o ustalonej licz-
```

```
bie przebiegów bez licznika */
say "Pętla o 5 przebiegach!"
do for 5
say "Halo!"
end
/* 3. Pętla bez końca. Aby jednak z niej wyjść, użyjemy licznika "i" i opuścimy pętlę przy pomocy "leave", gdy zostanie spełniony warunek i=6 */
```

```
i=0
say "Pętla bez końca!"
do forever
say "bez końca!"
i=i+1
if i=6 then leave
end
/* 4. Pętla "while". Przed każdym przebiegiem jest spr-
```

```

wzany podany warunek i w
zależności od niego pętla jest
wykonywana lub zakończona.
Jeśli warunek nie jest
spełniony przed pierwszym
przebiegiem pętli, pętla jest
ignorowana */
i=5
say "Pętla 'while'"
do while i=4
say "Ta pętla 'while' nie
zostanie wykonana"
end
do while i=5
say "Ta pętla 'while' będzie
wykonana 1 raz"
i=i-1
end
/* 5. Pętla "until". Tu jest
podany warunek; gdy jest
spełniony, pętla zostanie
przerwana. Oprócz tego
warunek jest sprawdzany po
każdym przebiegu pętli,
tak że pętla "until" jest
zawsze wykonana
przynajmniej raz */
i=4
say "Pętla 'until'"
do until i=4
say "Ta pętla 'until'
będzie wykonana 1 raz"
/* i to mimo tego, że
warunek jest spełniony */
end

```

Listing 11

```

/* Demonstracja leave i break
Najpierw break */
do
say "break-test1"
break /* Powoduje opuszczenie
bloku "do [...]" end"
dlatego następane polecenie
nie będzie wykonane */
say "break-test2"
end
/* Teraz leave */
do i=0 /* Pętla bez końca z
zmienną licznikową i */
say "leave-test1"
do for 5
leave i /* Powoduje opuszczenie
zewnętrznej pętli dlatego
drugie 'say' i tu nie będzie
wykonane */
end
say "leave-test2"
end

```

Listing 12

```

/* Test instrukcji "if" */
i=5
if i>4 then
say "i jest większe od 4"
else
say "i jest mniejsze lub
równie 4"
if i<6 then
say "i jest mniejsze od 6"

```

```

else do
say "to nie zostanie
wyprowadzone"
say "i to też nie"
end

```

Listing 13

```

/* Test instrukcji "select" */
i=5
select
when i>0 then say "i jest
dodatnie"
when i=0 then say "i = 0"
otherwise say "i jest
ujemne"
end

```

Listing 14

```

/* Test podprogramów */
test = "Halo!"
call procedura
say test
call podprogram
say test
exit
procedura:procedure
say test
test = "Podprogram!"
return
podprogram:
say test
test = "Podprogram!"
return

```

Listing 15

```

/* Demonstracja użycia
funkcji */
/* Wywołanie funkcji
przez podanie nazwy i
argumentu w nawiasie.
Gdy jest więcej
argumentów, oddziela
się je przecinkami */
i=silnia(6)
/* Wyprowadzenie
wyniku */
say "Silnia 6 wynosi i"
/* Drugim sposobem
wywołania funkcji
jest użycie "call".
Argument umieszcza
się po nazwie
funkcji, oddzielony
spacją. Dalsze
argumenty oddziela
się przecinkami. */
call silnia 7
/* Przy tym sposobie
wywołania funkcji
wynik znajdzie się
w zmiennej "result" */
say "Silnia 7 wynosi" result
/* Ponieważ funkcja
zwraca wynik, jej
wywołanie może
stanowić także
argument innej
funkcji lub
polecenia */
say "Silnia 5 wynosi"
silnia_rekurs(5)
exit /* Koniec
programu głównego */
/* Funkcja oblicza
silnię liczby
całkowitej
nieujemnej.
Używa lokalnych
zmiennych! */
silnia:procedure

```

```

arg i /* Argument
funkcji w zmiennej
i */
ret = 1
/* Ta pętla oblicza
iloczyn wszystkich
liczb całkowitych
od 1 do i */
do k=1 to i
ret=ret*k
end
/* Gdy po "return"
stoi argument,
zostaje zwrócony
jako wynik funkcji */
return ret /* Koniec
funkcji */
/* Ten podprogram
robi to samo.
Stanowi jednak
rekurencyjne
rozwiązanie
zadania i ma
Wam pokazać, że
tego rodzaju
rzeczy nie są w
AREXX'ie
niczym
niezwykłym */

```

```

silnia_rekurs:procedure
arg i
if i<=1 then ret=1 /*
Przerwanie
rekursji */
else ret=silnia_rekurs(i-1)*i
/* Według wzoru
rekurencyjnego
silnia(i)=silnia(i-1)*i */
return ret /* Koniec
funkcji */

```

Listing 16

```

/* Demo analizy
parametrów */
/* Argument, który
użytkownik podaje
przy wywołaniu
programu, zostaje
przydzielony
zmiennej "argument"
i wprowadzony */
parse arg argument
say argument
/* Teraz przykład
podziału danej
zmiennej. Zmienna
"test" zostaje
podzielona według
spacji */
test = "Chce mi się
pić"
parse var test a b c d
say a
say b
say c
say d

```

Listing 17

```

/* Jak uruchamia
się programy
Shellu? Prosty
program, który
kopiuje jeden
plik. Gdy brak
drugiego
parametru,
kopiowanie
odbywa się
automatycznie
do katalogu
aktualnego */
parse arg zdrojlo
przeznaczenie
if przeznaczenie = ""
then przeznaczenie = ""
address command 'copy'
zrodlo przeznaczenie

```

Listing 18

```

/* Nazwa programu:
bytes.rexx
Ustal sumę
długości plików
w bajtach w
podanym katalogu
i wszystkich
podkatalogach */

```

```

Markus Stoll, 31.5.91 */
parse arg pfad
/* Następująca
komenda wymusza
skok do etykiety
BREAK_C (i
zakończenie
programu),
gdzy użytkownik
wcisnie Ctrl-C */
signal ON BREAK_C
/* Funkcja
AREXX EXISTS()
sprawdza, czy
podana ścieżka
istnieje */
if EXISTS(pfad)
then ermittleumfang(pfad)
else say "Ścieżka
błędnie
podana!"
exit
BREAK_C:
say "Przerwanie"
exit
ermittleumfang:procedure
parse arg pfad
filename="RAM:count_temp"
/* Wywołanie
rozkazu DOS'u
"List" z
żądaną ścieżką
i skierowanie
wyniku do
pliku w
RAM-dysku */
bef = "list >" || filename
|| pfad || "" "NOHEAD
NO-DATES"
address command bef
/* Otwarcie
pliku z
listingiem
dla odczytu */
succ=OPEN(datei,filename,'R')
/* W
zasadzie
nie powinno
być błędu,
jeśli jednak
- wtedy
koniec
programu */
if succ = 0
then exit 10
dirs=0 /*
licznik
podkatalogów */
bytes=0 /*
ilość
bajtów w
danym
katalogu */
/* Pętla
czytająca
dane
linia po
linii i
wartościująca
aż do
napotkania
końca
pliku */
do until (EOF(datei))
zeile=READLN(datei)
/* Gdy
pierwszym
znakiem
jest ":"
->
komentarz
do
poprzedniego
pliku,
opuszczać!
(":" nie
może
przecież
zdarzyć
się w
nazwie
pliku) */
a=SUBSTR(zeile,1,1) /*
Pierwszy
znak
łańcucha */
if (COMPARE(a,":")>0)
then do call
parsezeile
if (COMPARE(name,"")>0)
then do
if (COMPARE(groesse,"Dir")=0)
then do
dir.dirs=name
dirs=dirs+1
end
else if (COMPARE(groesse,"empty")>0)
then
bytes=bytes+groesse
end
end
end

```

```

CLOSE(date1) /* zamknięcie pliku */
/* Usunięcie pliku tymczasowego */
address command "delete >nil:" filename
/* Przeszukanie podkatalogów */
if (dirs>0) then do
do i=0 to dirs-1
/* Rozszerzenie ścieżki na podkatalogi */
if (COMPARE(pfad,"")=0) then
temp=dir.i
else do
if (COMPARE(RIGHT(pfad,1),".")=0)

```

```

then
temp=pfad || dir.i
else temp=pfad || "/" || dir.i
end
/* Wywołanie rekurencyjne ze ścieżką podkatalogu */
bytes=bytes +
ermittleumfang(temp)
end
end
say pfad bytes
return bytes /* Wyprowadzenie długości dla rekurencyjnego wywołania */
/* Właściwie następna li-

```

```

nia powinna wystarczyć zamiast wywołania funkcji "parsezeile":
parse var zeile name groesse rest
Ponieważ jednak nazwy plików i katalogów mogą zawierać spacje, jest konieczne następujące, kosztowniejsze rozwiązanie */
parsezeile:
/* Chwyt: łańcuch przetwarzamy od końca dzięki inwersji przed "parse" i ponownej inwersji po */
/* reverse() zwraca podany łań-

```

```

cuch w odwróconym porządku (jakby przeczytany od tyłu) */
temp=REVERSE(zeile)
parse var temp bits groesse name
groesse=REVERSE(groesse)
/* Usuń jeszcze przy pomocy STRIP() spacje na końcu nazwy */
name=STRIP(REVERSE(name), "T")
return

```

AMIGA Magazin 8/1991
Tłum. Jan Pikul

SKŁADNIA ROZKAZÓW AREXXA

Marek Pampuch

■ C2B ([convert] c[haracter string into] b[inary string] — przekształć łańcuch znakowy w binarny) *** FUNKCJA

C2B(a\$)

gdzie: a\$ — to (jak łatwo się domyślić) łańcuch znakowy. Tworzy łańcuch binarny będący odpowiednikiem łańcucha a\$. UWAGA: Znaczenie słowa "Odpowiednik" należy potraktować w ten sposób, że każdemu znakowi łańcucha a\$ zostaje przypisany jego kod ASCII, a następnie kody te zostają przekształcone na postać binarną. Przykład:

SAY C2B('abc') (da w wyniku łańcuch '011000010110001001100011')

■ C2D ([convert] c[haracter string into] d[ecimal] — przekształć łańcuch znakowy w liczbę) *** FUNKCJA

C2D(a\$ [,n])

gdzie: a\$ — to łańcuch binarny lub heksadecymalny, zaś n — cyfra. Zamienia łańcuch a\$ na odpowiadający mu kod ASCII (podany cyfrą dziesiętną). Podczas konwersji dodawany jest znak bitu. Jeśli podasz parametr n, wówczas łańcuch a\$ zostanie potraktowany jako "słowo" n-bajtowe. W takim przypadku łańcuch zostanie odpowiednio zaokrąglony lub uzupełniony na początku zerami. Jeśli wartość łańcucha przekracza (dziesiętnie) 256, a nie użyjesz n lub n jest większe od 2, wówczas funkcja przyjmie wartość -1. Przykład:

SAY C2D('0020'X)

(da w wyniku 32),

SAY C2D('FFFF')

(da ten sam wynik), zaś

SAY C2D('FF01000'X,2)

(da w wyniku 256).

■ C2X ([convert] c[haracter string into] h[exadecimal string] — przekształć łańcuch znakowy w heksadecymalny) *** FUNKCJA

C2X(a\$)

gdzie: a\$ — to łańcuch znakowy.

Zamienia łańcuch znakowy w jego "odpowiednik" heksadecymalny. Przykład:

SAY C2X('abe')

(da w wyniku '616265').

■ CALL (wywołaj) *** INSTRUKCJA

CALL <a\$b\$> [w1 [,w2, ... , wn]] gdzie: a\$ — to zmienna

łańcuchowa, b\$ — łańcuch, zaś w1 ... wn — wyrażenia. Wywołuje funkcję wewnętrzną lub zewnętrzną o nazwie a\$ lub b\$. Użyte wyrażenia stają się parametrami wywołanej funkcji (tzn. wartościami, z jakimi funkcja będzie obliczana). Wartość przyjęta lub obliczona przez wywołaną funkcję zostanie podstawiona pod zmienną o nazwie RESULT. Przykład:

CALL center nazwa,l+4,'+

wywoła funkcję wewnętrzną CENTER (opisaną niżej), obliczy jej wartość z parametrami a\$="nazwa", l=l+4 i p\$="+". Jeśli przed wywołaniem funkcji zmienna l miała wartość na przykład 3 — wówczas pod zmienną RESULT zostanie podstawiony łańcuch '+nazwa+'.

■ CENTER (umieść w środku) *** FUNKCJA dopuszczalna jest również (w starszych wersjach AREXXA) nazwa CENTRE CENTER (a\$,l,[p\$]) gdzie: a\$ — to łańcuch, l — liczba, zaś p\$ — to łańcuch 1-znakowy. Tworzy nowy łańcuch o podanej długości (l), w którym umieszcza łańcuch a\$ na poniższych zasadach:

* Jeśli l jest większe niż długość łańcucha a\$, wówczas zostaną dopisane spacje, aż do uzyskania przez łańcuch wynikowy długości l — jedna na początku, pozostałe na końcu. (Jest to chyba niedoróbka w tej wersji, z jaką miałem do czynienia, bowiem nazwa sugerowałaby, że spacje powinny zostać umieszczone mniej więcej po połowie, przed i po łańcuchu a\$).

* Jeśli użyjesz parametru p\$, wówczas w miejscach, gdzie mają być spacje, zostanie umieszczony znak p\$,

* Jeśli l jest większe niż długość łańcucha, wówczas łańcuch wynikowy zostanie utworzony w ten sposób, że znak występujący na pierwszej pozycji łańcucha a\$ będzie środkowym znakiem łańcucha wynikowego, zaś z łańcucha a\$ zostanie usunięte tyle znaków, ile wynosi wartość parametru l. Przykład:

SAY CENTER('abc',6)

(da w wyniku łańcuch '_abc_'. UWAGA: Podkreślenia oznaczają tu spacje),

SAY CENTER('ABC',6,'=')

(da w wyniku '=ABC=='), zaś

SAY CENTER('abcdefg',3)

(da w wyniku 'bcd')

■ CLOSE (zamknij) *** FUNKCJA

CLOSE(a\$)

gdzie: a\$ — łańcuch alfanumeryczny.



Zamyka zbiór o podanej nazwie a\$ (uwaga: łańcuch a\$ musi być podany w apostrofach). Jeżeli zbiór o takiej nazwie nie został wcześniej otwarty, wówczas funkcja przyjmie wartość = 1. Przykład:

SAY CLOSE('moj_zbiór')

(zamknie zbiór o nazwie "moj_zbiór". Jeśli nie otworzyłeś go wcześniej — da w wyniku 1).

■ COMPRESS (zgnieć) *** FUNKCJA

COMPRESS (a\$ [,b\$])

gdzie: a\$,b\$ — łańcuchy.

Usuwa z łańcucha a\$ wszystkie spacje (zarówno wewnętrzne, jak i zewnętrzne). Jeśli użyjesz parametru b\$ — z łańcucha a\$ zostaną usunięte wszystkie znaki, jakie występują w zmiennej b\$. Przykład (w którym znak "_" oznacza spację):

COMPRESS('_Commodore_Amiga_')

(da w wyniku łańcuch 'CommodoreAmiga'),

COMPRESS('+12.34-56+12.82','+'.')

(da w wyniku '1234561282').

■ COMPARE (porównaj) *** FUNKCJA

COMPARE (a\$,b\$ [,c\$])

gdzie: a\$,b\$ — łańcuchy, c\$ — łańcuch 1-znakowy.

Porównuje łańcuchy a\$ i b\$ i podaje numer pierwszej pozycji, na której są różnice, lub 0 — gdy łańcuchy są identyczne. Jeśli łańcuchy mają różną długość, wówczas krótszy z nich zostaje uzupełniony na początku spacjami (lub znakiem podanym przez zmienną c\$) — tak, aby oba łańcuchy były tej samej długości. Przykład:

SAY COMPARE ('abcde','abcie')

(da w wyniku 4),

SAY COMPARE('abcd','abcd')

(da w wyniku 0), zaś

SAY COMPARE('abc','+++asm','+'.')

(da w wyniku 5)

■ COPIES (skopiuuj) *** FUNKCJA

COPIES (a\$,n)

gdzie:a\$ — łańcuch, n — liczba.

Tworzy nowy łańcuch

będący n-krotnym powtórzeniem łańcucha a\$. Jeśli podasz n=0 — wówczas w wyniku otrzymasz łańcuch pusty. Przykład:

SAY COPIES('bar',4)

(da w wyniku 'barbarbarbar').

■ D2C ([convert] d[ecimal into] 2[binary string] — przekształć liczbę w łańcuch dwójkowy) *** FUNKCJA

D2C (n)

gdzie: n — liczba.

Tworzy łańcuch, którego wartością jest binarny odpowiednik podanej liczby dziesiętnej n. Zera znajdujące się po konwersji na początku łańcucha wynikowego zostaną ucięte. Przykład: D2C(4) (da w wyniku '100').

■ D2X ([convert] d[ecimal into] x[adecimal string] — przekształć liczbę w łańcuch heksadecymalny) *** FUNKCJA

D2X (l [,n])

gdzie: l,n — liczby.

Zamienia liczbę dziesiętną l na łańcuch heksadecymalny. Jeśli użyjesz parametru n — łańcuch wynikowy zostanie przedstawiony jako liczba n-znakowa, na przykład:

D2X(10)

(da w wyniku '0A'X),

D2X(10,1)

(da w wyniku 'A'X).

■ DATATYPE (typ danej) *** FUNKCJA

ATATYPE (a\$ [,b\$])

gdzie: a\$ — łańcuch, b\$ — łańcuch 1-znakowy.

Jeśli nie użyjesz parametru b\$ — wówczas funkcja sprawdzi, jakim typem zmiennej jest łańcuch a\$ (jeśli jest zmienną liczbową — wtedy wynikiem będzie NUM, jeśli znakową — wynikiem będzie CHAR). Użycie parametru b\$ spowoduje testowanie łańcucha a\$ w zależności od znaku, jakim jest b\$. Masz do wyboru:

* b\$='A' — badanie, czy łańcuch a\$ składa się ze znaków alfanumerycznych (małe i duże litery alfabetu, cyfry 0-9).

* b\$='B' — badanie, czy a\$ jest łańcuchem binarnym,

* b\$='L' — sprawdzenie, czy zmienna a\$ składa się wyłącznie z małych liter alfabetu,

* b\$='M' — czy składa się wyłącznie z liter (małych i dużych),

* b\$='N' — czy składa się z cyfr,

* b\$='S' — czy zmienna a\$ zawiera wyłącznie prawidłowe (z punktu widzenia języka ARexx) symbole,

* b\$='U' — sprawdzi, czy zmienna a\$ składa się wyłącznie z dużych liter alfabetu,

* b\$='W' — zbada, czy a\$ jest liczbą całkowitą, zaś

* b\$='X' — czy jest łańcuchem heksadecymalnym.

Jeśli łańcuch a\$ spełnia warunki postawione przez test (określony przez b\$), funkcja przyjmuje wartość true(1), w przeciwnym wypadku — false (0). Przykład:

SAY DATATYPE('123')

(da w wyniku NUM),

SAY DATATYPE('1AF2','X')

(wynikiem będzie 1), zaś z

SAY DATATYPE('AbCdEf','U')

(wyjdzie 0).

■ DATE (data) *** FUNKCJA

DATE ([a\$],[d],[b\$])

gdzie: a\$,b\$,d — łańcuchy, d-liczba całkowita.

Ustawia aktualną datę w formacie zależnym od podanych parametrów. Parametr a\$ może przyjmować następujące wartości:

B (beginning — początek) — liczba dni od 1.01.0001,

C (century — wiek) — liczba dni od 1.01.1900,

D — liczba dni od 1.01 obecnego roku,

E — data w formacie DD/MM/RR (gdzie: DD — dzień (liczba), MM-miesiąc (liczba), RR-rok),

I — liczba dni w formacie wewnętrznym (liczba),

J — data w formacie RRRMM, gdzie MMM to trzyliterowy skrót angielskiej nazwy miesiąca (na przykład Jan — January — styczeń),

M — pełna angielska nazwa miesiąca (na przykład February — luty),

N — data w formie DD MMM RRR (ustawiona domyślnie, gdy opuścisz parametr a\$),

O — data w formie RR/MM/DD,

S — data w formie RRRR MM DD,

U (US Standard) — data w formacie MM/DD/RR (jak na IBM),

W (weekday) — angielska nazwa dnia tygodnia.

Parametr d określa, czy data ma być podana w postaci prostej (rok na końcu — ustawione defaultowo), czy "sortowanej" (rok na początku). Parametr b\$ — ma znaczenie podobne jak a\$ — z tym, że może przybierać tylko 2 wartości (I lub S). Dla daty 10 luty 1991:

SAY DATE()(da w wyniku 10 Feb 1991),

SAY DATE('M')

(wynikiem jest: February),

SAY DATE('S')(w wyniku otrzymasz: 1991021.)

UWAGA: Mimo że dziś mamy (założmy) 10.2.91, w mojej Amidze data systemowa (bo nie mam zegara) jest akurat: 20.4.1988. Zatem:

SAY DATE('S',DATE('I')+21) (daje '19890609'), zaś

SAY DATE('W',19890609,'S')

(powoduje wynik:'Friday').

■ DELSTR (del[ete] str[ing] — usuń łańcuch) *** FUNKCJA

DELSTR (a\$,n [,l])

gdzie: a\$ — łańcuch, n,l — liczby całkowite.

Usuwa z łańcucha a\$ jego fragment zaczynający się od n-tego znaku i o długości l znaków. Jeśli opuścisz parametr l — wówczas defaultowo zostanie usunięta reszta łańcucha (tzn. od

pozycji n do końca). Przykładowo:

```
SAY DELSTR('123456',2,3)
```

(da w wyniku łańcuch '156').

■ **DELWORD** (del[ete] word [from string] — usuń słowo z łańcucha) *** FUNKCJA

```
DELWORD (a$,n [,l])
```

gdzie: a\$ — łańcuch, n,l — liczby całkowite.

Usuwa z łańcucha a\$ — l słów (za słowo jest uważany fragment łańcucha, po którym następuje spacja). Usuwanie rozpoczyna się od n-tego z kolei słowa. Jeśli opuścisz parametr l — zostaną usunięte wszystkie słowa poczynając od n-tego słowa do końca łańcucha. UWAGA: Łańcuch wynikowy zawiera spację, która następuje po ostatnim nie usuniętym słowie. Przykład:

```
SAY DELWORD('Opowiedz mi ładna bajke',2,2)
```

(wypisze na ekranie "Opowiedz bajke"),

```
SAY DELWORD('001 002 003 004 005',3)
```

(da w wyniku łańcuch '001 002')

■ **DIGITS** ([return] digits [number set] — podaj liczbę ustawionych miejsc dziesiętnych (tu: precyzję) *** FUNKCJA

```
DIGITS ()
```

Podaje precyzję, jaka została ostatnio ustawiona instrukcją **NUMERIC**. Jeśli jest to (przykładowo) **NUMERIC DIGITS 8**, wówczas

```
SAY DIGITS()
```

(da w wyniku 8).

■ **DO** (zrób) *** INSTRUKCJA

```
DO [n1=w1 [TO w2] [BY w3] [FOR w4] [FOREVER] [WHILE w5 | UNTIL w6]
```

Znaczenie parametrów: n1 — nazwa zmiennej, w1 ... w6 — wyrażenia. Jest to odpowiednik Basicowego **FOR**. Instrukcja **DO** otwiera grupę rozkazów, które mają być wykonywane wielokrotnie (czyli "pętli"). Każdy blok instrukcji otwartych przez **DO** — musi kończyć się instrukcją **END**. Wszystkie rozkazy, zawarte pomiędzy **DO** i **END**, są w **ARexx**'ie nazywane zakresem działania instrukcji **DO**. Nie przestrasz się zatem, jeśli napotkasz gdzieś taką nazwę. Wszystkie parametry w **DO** są opcjonalne (tzn. możesz je opuścić). Jeśli opuścisz wszystkie parametry — wówczas instrukcje zawarte w bloku **DO ... END** zostaną wykonane tylko raz. Parametr n1 jest zmienną licznikową przyjmującą wartość wyrażenia w1. Jeśli używasz jakiegokolwiek opcji (poza **FOREVER**), wówczas zmienna licznikowa musi być pierwszym wyrażeniem następującym bezpośrednio po instrukcji **DO**.

Wyrażenie w2 (po słowie **TO**) określa wartość zmiennej licznikowej, przy której nastąpi zakończenie działania bloku **DO ... END**.

Wyrażenie w3 (po słowie **BY**) — określa przyrost zmiennej licznikowej i może przyjmować dowolne wartości numeryczne. Jeśli opuścisz parametr **BY** — wówczas wartość w3 zostanie domyślnie ustawiona na 1.

Wyrażenie w4 (po słowie **FOR**) musi dawać wynik, będący liczbą całkowitą określającą minimalną ilość iteracji, jaka ma być wykonana. Opcji **FOR** można używać razem z opcjami **BY** i **TO**.

Użycie słowa kluczowego **FOREVER** — spowoduje wykonywanie instrukcji do momentu napotkania w bloku **DO ... END** instrukcji **LEAVE** lub **BREAK**. UWAGA: W tym przypadku nie należy używać zmiennej licznikowej. Z tego wynika wniosek, że **FOREVER** powoduje wykonywanie pętli bez nadawania indeksów zmiennym (na przykład, jeśli chcesz zatrzymać przez chwilę program). Identyczny efekt jak opcja **FOREVER** da instrukcja:

```
DO i=1
```

Wyrażenie w5 (po słowie **WHILE**) jest obliczane przy każdej nowej iteracji. Wynik wyrażenia musi być równy 1 (wówczas następuje kolejna iteracja) lub 0 — wówczas "pętla" kończy swoje działanie.

Na identycznych zasadach (tyle że "odwrotnie" — tj: 0-dalsza iteracja, 1-koniec) "działa" wyrażenie w6 w opcji **UNTIL**. Przykład:

```
DO i=1 TO limit FOR 5 WHILE time<10
```

```
y.i=i*time
```

```
END
```

teraz (dla porównania) "robiąca to samo" pętla w **BASIC**-u:

```
10 i=1
```

```
20 y(i)=i*time:if time>10 then end
```

```
if i<5 then 30
```

```
if i=limit then end
```

```
30 i=i+1:goto 20
```

I co jest prostsze? Jeśli jeszcze nie jesteś przekonany, spróbuj "uprościć" sobie program w **BASIC**-u przez zastosowanie pętli **FOR ... NEXT**.

■ **DROP** (wypadnij) *** INSTRUKCJA

```
DROP a [b ... z]
```

gdzie: a-z — symbole zmiennych dowolnego typu.

Jako wartość zmiennej zostanie podstawiona nazwa zmiennej. Jeśli użyjesz nazwy zmiennej rdzeniowej — wówczas operacja **DROP** zadziała na wszystkich — możliwych do uzyskania z tej zmiennej — zmiennych złożonych. Czynność wykonywana przez funkcję **DROP** nazywana bywa czasem "antyinicjalizacją". Jeśli usiłujesz "zDROPić" zmienną, którą już wcześniej tak potraktowałeś — wówczas nic się nie stanie. Na przykład:

```
a=1;b=12
```

```
SAY a b
```

(tu zostanie wydrukowane 1 i 12), ale jeśli użyjesz

```
DROP a b
```

```
SAY a b
```

(to zostanie wydrukowane A i B).

■ **ECHO** *** INSTRUKCJA

```
ECHO w1
```

Parametr w1 — to wyrażenie.

Wyświetla

wynik wyrażenia na ekranie (tak samo jak **SAY**). Uwaga: Jeśli wyrażenie jest łańcuchem — wówczas musi być tu podane w cudzysłowie. Przykładowo:

```
ECHO "AMIGA 500"
```

(da ten sam efekt co

```
SAY 'AMIGA 500'
```

a będzie to wypisanie na ekranie najmilej brzmiącej na świecie nazwy.)

■ **ELSE** (w innym przypadku) *** INSTRUKCJA

```
ELSE [:] [w1]
```

gdzie: w1 — to wyrażenie warunkowe.

Instrukcja (podobnie jak w **BASIC**-u) pozwala na wybór różnych możliwości w instrukcji **IF**. UWAGA: Nie może być użyta poza zakresem działania instrukcji **IF** i musi występować po instrukcji **THEN**. Jeśli zagnieżdżysz w sobie kilka **IF**-ów, wówczas **ELSE** działa w zakresie najbliższego poprzedniego **IF**. W przypadku zagnieżdżenia, nie wolno stosować "pustego" **ELSE** (zamiast tego musisz użyć instrukcji **NOP**) ani też **ELSE** ze średnikiem (**ELSE;**, które jest równoważne "pustemu" **ELSE**). Przykład:

```
IF i<>2 THEN SAY 'chyba nie'
```

```
ELSE SAY 'chyba tak'
```

■ **END** (koniec) *** INSTRUKCJA

```
END [n]
```

Parametr n — jest symbolem zmiennej licznikowej w instrukcji **DO**. Instrukcja **END** kończy zakres działania instrukcji **DO** lub **SELECT**. Zmiennej opcjonalnej n użyj wtedy, jeśli masz w programie wiele instrukcji **DO** i chcesz być pewny, że **END** odnosi się do właściwego **DO** (jeśli nazwy symboli się nie zgadzają, wówczas wystąpi błąd). Przykład:

```
DO i=1 TO 11 BY 2
```

```
SAY i
```

```
END i
```

(na ekranie zostaną wydrukowane cyfry 1,3,5,7,9,11).

■ **EOF** (e[nd] o[f] f[ile] — koniec zbioru) *** FUNKCJA

```
EOF (a$)
```

gdzie: a\$ — to łańcuch znakowy.

Funkcja sprawdza, czy zbiór o nazwie a\$ (UWAGA: Tu nazwa jest podana bez apostrofów) ma ustawiony znacznik końca zbioru (end of file). Jeśli tak — wówczas funkcja przyjmuje

wartość "true" (1), w przeciwnym wypadku "false" (0). Przykład: SAY EOF(moj_zbior)

(Jeśli zbiór "moj_zbior" ma znacznik końca, wówczas na ekranie pojawi się 1, jeśli zaś go nie ma, wtedy pojawi się 0).

■ **ERRORTEXT** ([return] error [message] text — podaj tekst komunikatu o błędzie) *** **FUNKCJA**

ERRORTEXT (n)

gdzie: n — liczba z zakresu 1 — 48.

Funkcja przyjmuje jako wartość informację o błędzie, którego kod wynosi n. Jeśli liczba n jest niecałkowita lub spoza zakresu — wówczas zostanie utworzony łańcuch pusty. ARexx informuje o następujących błędach:

- * kod=1 Program not found — nie znaleziono programu o podanej nazwie, lub nie jest to program ARexxa.
- * 2 — Execution halted — program został zatrzymany — albo z klawiatury (jeśli naciśnąłeś jednocześnie klawisze [Ctrl] i [C]), lub z użyciem przerwania programowego. UWAGA: Z tego błędu możesz wyjść, jeżeli wcześniej uaktywniłeś przerwanie typu HALT.
- * 3 — Insufficient memory — brak wystarczającej ilości pamięci do wykonania programu.
- * 4 — Invalid character — w programie użyłeś nie zdefiniowanego wcześniej znaku niestandardowego (na przykład o kodzie 281, spoza zakresu ASCII). Aby zdefiniować taki znak — musisz przed pierwszym użyciem przedstawić go w postaci łańcucha binarnego lub heksadecymalnego.
- * 5 — Unmatched quote — Otworzyłeś cudzysłów lub apostrof i zapomniałeś go zamknąć.
- * 6 — Unterminated comment — Otworzyłeś komentarz (znakami / *), ale nie zamknąłeś go (przez */).
- * 7 — Clause too long — Użyte przez Ciebie wyrażenie jest zbyt długie, aby zmieściło się w buforze. Musisz podzielić je na kilka krótszych.
- * 8 — Invalid token — w wyrażeniu znajduje się nieprawidłowy (z punktu widzenia ARexxa) znak lub operator.
- * 9 — Symbol or string too long — użyta zmienna lub łańcuch ma zbyt długą nazwę.
- * 10 — Invalid message packet — próbowałeś przestać nieprawidłowy rozkaz do programu rezydentnego ARexx'a.
- * 11 — Command string error — błąd w nazwie rozkazu.
- * 12 — Error return from function — Niewłaściwe parametry funkcji zewnętrznej spowodowały generację kodu błędu funkcji.
- * 13 — Host environment not found — brak portu komunikatów odpowiadającego obecnemu adresowi zarządzającemu lub adres zarządzający nie jest uaktywniony.
- * 14 — Requested library not found — Próbowałeś otworzyć bibliotekę o nazwie (lub numerze wersji) niezgodnej z tą, jaką zapisałeś w spisie bibliotek (Library List), lub biblioteki o podanej nazwie nie ma na dysku.
- * 15 — Function not found — Wywołałeś funkcję, która:
 - nie jest wbudowana,
 - nie jest zdefiniowana jako program zewnętrzny lub
 - nie ma jej w spisie bibliotek.
- * 16 — Function did not return value — wywołałeś funkcję, a ta funkcja złośliwie nie przyjęła żadnej wartości (bo na przykład nie spodobały się jej Twoje parametry, choć na oko ilość i jakość parametrów się zgadza).
- * 17 — Wrong number of arguments — tu również wywołana funkcja zastrajkowała, gdyż podałeś jej zbyt małą lub zbyt dużą liczbę parametrów (w stosunku do liczby wymaganej).
- * 18 — Invalid argument to function — Opuściłeś argument funkcji, który jest niezbędny do prawidłowego działania funkcji lub podałeś go w sposób niewłaściwy.
- * 19 — Invalid PROCEDURE — Użyłeś instrukcji PROCEDURE w niewłaściwym kontekście. Błąd ten występuje również, gdy w PROCEDURE wywołujesz nie uaktywnioną funkcję wewnętrzną lub PROCEDURE zostało wcześniej umieszczone w aktualnym środowisku pamięciowym.
- * 20 — Unexpected THEN or WHEN — użyłeś jednej z tych dwu instrukcji w niewłaściwym kontekście (WHEN można używać tylko w

zakresie SELECT lub DO, THEN może występować po IF lub WHEN).

- * 21 — Unexpected ELSE or OTHERWISE — użyłeś którejś z tych instrukcji w złym kontekście (OTHERWISE może być użyte tylko w zakresie SELECT, ELSE można użyć tylko po IF).
- * 22 — Unexpected BREAK, LEAVE or ITERATE — użyłeś którejś z tych instrukcji poza zakresem instrukcji DO.
- * 23 — Invalid statement in SELECT — W zakresie instrukcji SELECT mogą znajdować się tylko WHEN, THEN i OTHERWISE oraz wyrażenia warunkowe związane z tymi instrukcjami.
- * 24 — Missing or multiple THEN — Zapomniałeś umieścić THEN po WHEN lub IF albo użyłeś zbyt dużo THEN po pojedynczym IF (WHEN).
- * 25 — Missing OTHERWISE — jeśli żaden z warunków postawionych przez WHEN nie jest spełniony — na końcu MUSI występować instrukcja OTHERWISE.
- * 26 — Missing or unexpected END — nie zamknąłeś zakresu instrukcji DO(SELECT) przez END lub użyłeś END poza zakresem wyznaczonym przez tę instrukcję (na przykład po kolejnym DO).
- * 27 — Symbol mismatch — symbol określający zmienną użyty w instrukcjach END, ITERATE lub LEAVE nie zgadza się z symbolem będącym zmienną licznikową poprzedzającej je instrukcji DO, lub dwa zakresy wyznaczone przez DO zachodzą na siebie.
- * 28 — Invalid DO syntax — jeżeli użyłeś parametrów TO lub BY w instrukcji DO — muszą być one poprzedzone określeniem zmiennej indeksującej.
- * 29 — Incomplete IF or SELECT — W instrukcji IF lub SELECT nie umieściłeś wszystkich wymaganych przez te instrukcje parametrów (na przykład wyrażen warunkowych po słowach THEN, ELSE, OTHERWISE).
- * 30 — Label not found — brak w programie etykiety użytej w instrukcji SIGNAL lub etykiety, do której odwołujesz się w programie bezpośrednio przez przerwanie. UWAGA: Błąd nie wystąpi, jeśli etykieta zdefiniowana jest dynamicznie instrukcją INTERPRET lub przez interaktywne wprowadzenie z klawiatury.
- * 31 — Symbol expected — w miejscu, gdzie w programie powinna znajdować się nazwa zmiennej (na przykład w instrukcjach END, LEAVE, UPPER itd) — umieściłeś inny znak lub pominąłeś tę nazwę.
- * 32 — Symbol or string expected — W miejscu, gdzie może występować wyłącznie nazwa zmiennej lub łańcuch — umieściłeś inny znak.
- * 33 — Invalid keyword — błędnie wpisałeś słowo kluczowe (czyli to, co w opisie jest dużymi literami).
- * 34 — Required keyword missing — Zapomniałeś umieścić w instrukcji słowa kluczowego, które musi występować (na przykład słowa kluczowe dotyczące obsługi przerwania typu SYNTAX itd. w instrukcji SIGNAL).
- * 35 — Extraneous character — użyłeś niepotrzebnych (nadmiarowych) znaków na końcu wyrażenia.
- * 36 — Keyword conflict — Użyłeś dwóch wzajemnie wykluczających się słów kluczowych (czyli takich, które w opisie rozdzielone są znakiem |) lub zastosowałeś dwa identyczne słowa kluczowe w tej samej instrukcji.
- * 37 — Invalid template — szablon zamieszczony w instrukcjach ARG, PARSE lub PULL został niewłaściwie napisany.
- * 38 — Invalid TRACE request — słowo kluczowe w instrukcji TRACE lub argument w funkcji wewnętrznej TRACE są nieprawidłowe.
- * 39 — Uninitialized variable — przy uaktywnionych przerwaniami typu NOVALUE próbowałeś wywołać zmienną niezainicjalizowaną.
- * 40 — Invalid variable name — próbowałeś podstawić nową wartość pod nazwę oznaczającą stałą.
- * 41 — Invalid expression — podczas obliczania wartości wyrażenia nastąpił błąd (na przykład dzielenie przez 0). Błąd wystąpi także przy niewłaściwej liczbie operandów w wyrażeniu lub przy nadmiarowych znakach w tym wyrażeniu
- * 42 — Unbalanced parentheses — nie zamknąłeś otwartego nawiasu.
- * 43 — Nesting limit exceeded — użyłeś za dużej liczby zagnieżdżonych instrukcji DO (więcej niż 10) lub "podprogramów" w wyrażeniu (więcej niż 32). Możesz usunąć błąd zamieniając "pętlę"

DO na iterację lub rozdzielając wyrażenie na dwa lub więcej wyrażeni.

* 44 — Invalid expression result — obliczona wartość wyrażenia jest nieprawidłowa (na przykład skok lub wartość końcowa zmiennej licznikowej w instrukcji DO jest nienumericzna).

* 45 — Expression required — opuściłeś wymagane w danym kontekście wyrażenie (na przykład, jeśli w instrukcji SIGNAL opuścisz słowo kluczowe ON (lub OFF), wówczas musisz użyć wyrażenia).

* 46 — Boolean value not 0 or 1 — wyrażenie logiczne przyjęło inną wartość niż 0 lub 1.

* 47 — Arithmetic conversion error — Użyłeś operandu nienumericznego w wyrażeniu numerycznym lub błędnie wpisałeś łańcuch binarny czy heksadecymalny.

* 48 — Invalid operand — użyłeś nieprawidłowego operandu (na przykład dzielenie przez 0) lub w funkcji wykładniczej użyłeś wykładnika niecałkowitego. UWAGA: Każdy błąd powoduje generację Amigowskiego kodu zwrotnego (z wszelkimi wynikającymi z tego konsekwencjami). Przeważnie kodem zwrotnym jest 10, jedynie błędy o kodzie ARexx'a 3i43 generują Amigowski kod zwrotny = 20, zaś błąd ARexx'a o kodzie 1 generuje Amigowski kod zwrotny = 5. Przykład:

SAY ERRORETEXT(42)

(spowoduje wypisanie komunikatu "Unbalanced parentheses").

■ EXISTS ([file] exists — czy zbiór istnieje) *** FUNKCJA EXISTS (a\$)

gdzie: a\$ — łańcuch (tu podany w apostrofach).

Sprawdza, czy istnieje zewnętrzny zbiór o nazwie a\$. Łańcuch może zawierać ścieżkę. W wyniku otrzymasz wartość = 1 (gdy zbiór istnieje) lub = 0 (gdy go nie ma). Przykład:

SAY EXISTS('df1:c/ed')

(da w wyniku na przykład 1 — jeśli na dyskietce w stacji df1: w katalogu c jest zbiór o nazwie "ed").

■ EXIT (wyjście) *** INSTRUKCJA

EXIT [w1]

Parametr w1 jest wyrażeniem.

Instrukcja przerywa wykonywanie programu. Możesz ją umieścić w dowolnym miejscu programu. Jeżeli użyłeś parametru w1 — wówczas wartość wyrażenia zostanie obliczona i przesłana do miejsca w programie zewnętrznym, z którego nastąpiło wywołanie programu przerwane przez EXIT. Przykład:

EXIT

(przerywa wykonanie programu i powraca bez obliczonej wartości wyrażenia).

EXIT 10

(przerywa wykonanie programu i generuje kod zwrotny 10).

■ EXPORT (wyślij) *** FUNKCJA

EXPORT (a\$ [,b\$] [,l] [,c\$])

Znaczenie parametrów jest następujące: a\$,b\$ — łańcuchy, c\$ — łańcuch 1-znakowy, l — liczba całkowita. Funkcja EXPORT kopiuje łańcuch b\$ w obszar pamięci o adresie a\$ (który musi być 4-bitowym łańcuchem binarnym lub heksadecymalnym). Jeśli użyjesz parametru l — wówczas zostanie skopiowane l pierwszych znaków łańcucha b\$ (domyślnie przyjęta jest całkowita długość łańcucha b\$). Jeśli l jest większe niż długość łańcucha, wówczas łańcuch zostanie uzupełniony na początku zerami, a jeśli użyłeś parametru c\$ — znakiem, jaki występuje w c\$. Jednocześnie funkcja przyjmie wartość = l. UWAGA: Początkującym "AReksiom" radzę NIE EKSPERYMENTOWAĆ z tą funkcją, jako że kopiowany łańcuch b\$ zapisuje się na tym, co już jest w pamięci pod adresem a\$. Od momentu użycia funkcji nie wolno przelączać dostępu. Zabawa w EXPORT-erów może zakończyć się zawieszeniem komputera!!!. Przykład:

co=EXPORT('0004 0000'X,'AMIGA')

(skopiuje słowo "Amiga" do komórki o adresie 40000, a zmienna co przyjmie wartość 5).

■ FORM ([return] form — podaj formę) *** FUNKCJA FORM()

Podaje nazwę aktualnie ustawionego formatu zapisu cyfr. Przykład:

NUMERIC FORM SCIENTIFIC

SAY FORM()

(wypisze na ekranie słowo SCIENTIFIC (tu notacja "naukowa"))

■ FIND (znajdź) *** FUNKCJA

FIND (a\$,b\$), gdzie: a\$,b\$ — to łańcuchy.

Określa pozycję zestawu słów b\$ w większym zestawie słów a\$ i przyjmuje wartość równą miejscu występowania pierwszego słowa łańcucha b\$ w łańcuchu a\$. Jeśli nie znaleziono danego zestawu słów b\$ — wówczas funkcja przyjmie wartość = 0. Przykład:

SAY FIND ('Najlepszym komputerem jest Commodore Amiga', 'jest Commodore') (da w wyniku 3).

■ FREESPACE (wolne miejsce [tu: ilość pamięci]) *** FUNKCJA

FREESPACE (a\$,b)

Znaczenie parametrów: a\$ — łańcuch, b — liczba.

Przekazuje blok pamięci rozpoczynający się od adresu a\$ (łańcuch 4-bajtowy) i mający długość l do wewnętrznego bufora interpretera. Jeśli opuścisz argumenty [UWAGA: składnia w tym przypadku musi być: FREESPACE()] — wówczas funkcja poda wielkość pamięci wewnętrznego bufora interpretera. Nie jest konieczne używanie tej funkcji na początku programu, ponieważ wówczas alokacja zostaje wykonana automatycznie, należy jednak z niej korzystać wtedy, gdy zaczynasz mieć problemy z pamięcią w programie. Funkcja przyjmuje wartość 1 — gdy operacja przebiegnie pomyślnie, lub wartość 0 — gdy jest niemożliwa. UWAGA: Wygodnie jest uzyskać adres pierwszej wolnej komórki bufora interpretera za pomocą opisanej dalej funkcji GETSPACE. Przykład:

SAY FREESPACE ('0004 2000'X,32)

(przetransferuje 32 bajty pamięci do komórki 42000 i (gdy wszystko przebiegnie pomyślnie) przyjmie wartość równą 1.

■ FUZZ (zakłócenia) *** FUNKCJA

FUZZ()

Podaje aktualnie ustawiony poziom zakłóceń numerycznych. Przykład:

NUMERIC FUZZ 3

SAY FUZZ()

(da w wyniku 3).

■ GETCLIP (get [from] Clip[board list] — pobierz z listy bufora) *** FUNKCJA

GETCLIP(a\$)

gdzie: a\$ — to łańcuch.

Szuka na liście bufora (Clip List) danej odpowiadającej nazwie podanej łańcuchem a\$ (tu a\$ musi być umieszczone w apostrofach) i przyjmuje wartość tej danej (lub 0, gdy nic nie zostało znalezione). Jeśli w buforze znajduje się na przykład wyrażenie 'liczba' o wartości równej 'PI=3.142' i użyjesz SAY GETCLIP ('liczba'), wówczas w wyniku otrzymasz: PI=3.14.

■ GETSPACE (pobierz [adres pamięci z wewnętrznego bufora interpretera]) *** FUNKCJA

GETSPACE(l) gdzie: l — to liczba całkowita.

Alokuje blok pamięci o podanej długości l w wewnętrznym buforze interpretera i przyjmuje jako wartość adres pierwszej wolnej komórki w tym buforze w postaci 4-bajtowego łańcucha. Nie zaleca się jej stosowania w programach zewnętrznych obsługiwanych przez ARexx. Przykład:

SAY C2X(getspace(32))

(przealokuje 32-bajtowy blok pamięci, przekształci otrzymany adres na liczbę heksadecymalną i wypisze na ekranie '0003BF40'X [w przypadku gdy jest to pierwsze użycie funkcji od początku programu])

■ HASH (podaj kod pomyłkowy [niedozwolony — tzw. op-code]) *** FUNKCJA

HASH (a\$) gdzie: a\$ — to łańcuch 1-znakowy.

Podaje tzw. op-code dla znaku a\$ (w postaci cyfry dziesiętnej) i aktualizuje ten op-code. Przykład:

SAY HASH('1')

(wyświetli na ekranie 49).

■ IF (jeśli) *** INSTRUKCJA

IF w1 THEN [;] [w2]

Znaczenie parametrów: w1 — wyrażenie, w2 — wyrażenie warunkowe. Pozwala na sterowanie działaniem programu w zależności od tego, czy wyrażenie w1 jest spełnione (tzn. przyjmuje wartość logiczną "true" (1), czy nie ("false"(0)). W przypadku "true" zostanie wykonane wyrażenie w2, w przypadku "false" program przejdzie do następnego wiersza, który może zaczynać się instrukcją ELSE. Przykład w instrukcji ELSE.

■ IMPORT (przyślij, tu: pobierz) *** INSTRUKCJA

IMPORT(a\$ [,l])

gdzie: a\$ — łańcuch 4-bajtowy, l — liczba całkowita.

Wynikiem działania instrukcji jest łańcuch utworzony przez skopiowanie z pamięci obszaru, rozpoczynającego się w komórce o adresie a\$ i o długości l. Jeżeli pominiemy l — zawartość pamięci będzie dopisywana do łańcucha wynikowego aż do momentu napotkania bajtu zerowego. Przykład: SAY IMPORT('0004 0000'X,8)

(da w wyniku zawartość 8 komórek pamięci, począwszy od 40000 (na przykład FF FF FF 00 3A BE FF FF), zaś

SAY IMPORT('0004 0000'X)

(da w wyniku łańcuch 'FF FF FF').

■ INDEX *** FUNKCJA

INDEX(a\$,b\$ [,l])

Parametrami są tu: a\$,b\$ — łańcuchy, l — liczba całkowita. Szuka miejsca występowania wzorca b\$ w łańcuchu a\$ — rozpoczynając szukanie od l-tej pozycji łańcucha a\$ (gdy opuścisz l, wówczas przeszukiwanie następuje od początku łańcucha. Jako wartość przyjmuje pozycję znalezionej wzorca w łańcuchu a\$ (licząc od początku łańcucha a\$) lub wartość 0, gdy dany wzorec w łańcuchu nie występuje. UWAGA: W tej funkcji — łańcuchy są w cudzysłowie. Przykład:

SAY INDEX("1234567", "23")

(da w wyniku 2),

SAY INDEX("1234567", "890")

(da w wyniku 0), zaś

SAY INDEX("123123123", "23", 3)

(da w wyniku 5).

■ INSERT (wsuń) *** FUNKCJA

INSERT (a\$,b\$ [,s] [,l] [,c\$])

gdzie: a\$, b\$ — to łańcuchy, c\$ — łańcuch 1-znakowy, l, s — liczby całkowite.

Wsusza łańcuch a\$ w łańcuch b\$ poczynając od pozycji s+1 (jeśli opuścisz s — wówczas jest dopisywany na początku, jeśli s jest większe niż długość łańcucha b\$, wówczas na początku łańcucha wynikowego występuje łańcuch b\$, dalej znajduje się tyle spacji, ile wynosi różnica między s i długością łańcucha b\$, zaś na końcu występuje łańcuch a\$. Jeśli użyjesz parametru c\$, wówczas w miejscu spacji zostaną umieszczone znaki, jakie zawiera zmienna c\$. Parametr l podaje, jaką długość ma mieć łańcuch a\$ jako część łańcucha wynikowego. Przykład:

SAY INSERT('ab', '12345')

(da w wyniku łańcuch 'ab12345'), natomiast

SAY INSERT('abc', '12', 3, 5, '*')

utworzy łańcuch '12*abc**'

■ INTERPRET (zinterpretuj) *** INSTRUKCJA

INTERPRET w1

gdzie: w1 — to wyrażenie. W wyniku działania instrukcji INTERPRET — wyrażenie w1 zostanie wykonane, a jego wynik zostanie potraktowany jako instrukcja programowa. Jako wyrażenie można użyć także dowolnej instrukcji lub bloku instrukcji (typu DO ... END). Instrukcja uaktywnia swój zakres kontrolny, z którego można wyjść instrukcjami LEAVE lub ITERATE (z zachowaniem efektu działania INTERPRET) lub przez BREAK (z likwidacją tego efektu). Instrukcja pozwala na dynamiczne konstruowanie i testowanie programów (dzięki traktowaniu rozkazu jako zmiennej) lub na użycie fragmentów programu jako argumentów instrukcji. Przykład:

naz = 'say'

(tu podstawiamy instrukcję jako zmienną naz)

INTERPRET naz Amiga500

(efektem działania będzie wykonanie rozkazu SAY 'Amiga500', czyli wypisanie tekstu Amiga500 na ekranie).

■ ITERATE (ziteruj) *** INSTRUKCJA

ITERATE [n]

gdzie: n — to nazwa zmiennej.

Instrukcja ta przerywa działanie iteracji w pętli DO ... END i zaczyna iterację od początku. Jeśli w pętli nie umieszisz rozkazu LEAVE, wówczas przy 11 kolejnej iteracji może wystąpić błąd. Jeśli użyjesz nazwy zmiennej — musi być ona taka sama jak w odpowiadającej instrukcji DO. Założmy, że j=3. Jeśli wpiszesz poniższy przykładowy programik:

DO i=1 to 7 IF i=j THEN ITERATE i

ELSE SAY i

LEAVE

END

(wówczas dziesięciokrotnie wydrukowana zostanie para liczb 1,2 — po czym nastąpi wyskok z pętli).

■ LASTPOS (last pos[ition] — ostatnia pozycja) *** FUNKCJA

LASTPOS (a\$,b\$ [,s])

Znaczenie parametrów: a\$ — łańcuch 1-znakowy, b\$ — łańcuch, s — liczba całkowita.

Szuka wzorca a\$ w łańcuchu b\$, rozpoczynając od pozycji s. Różnica między LASTPOS i FIND polega na tym, że przeszukiwanie odbywa się w tył. Jeśli opuścisz parametr s, wówczas szukanie rozpocznie się od ostatniego znaku łańcucha b\$. Funkcja przyjmuje wartość pozycji, na której znajduje się wzorec, lub 0, gdy wzorec nie został znaleziony. Przykład:

SAY LASTPOS('2', '12345')

(da w wyniku 2),

SAY LASTPOS('2', '123123123')

(da w wyniku 8), zaś

SAY LASTPOS('2', '123234', 3)

(da w wyniku 2).

LEAVE (pozostaw) *** INSTRUKCJA

LEAVE [n]

gdzie: n — to nazwa zmiennej.

Pozwala na natychmiastowe wyjście z zakresu instrukcji DO ... END lub ITERATE. Jeśli użyjesz nazwy zmiennej, wówczas musi ona być taka sama jak w odpowiednim DO. W przeciwnym razie wystąpi błąd. Wystąpi on również wtedy, gdy użyjesz LEAVE poza zakresem DO ... END. Przykład:

DO i=1 TO limit

IF i>10 THEN LEAVE i

END

(pętla DO ... END zostanie wykonana 10 razy, po czym nastąpi wyskok z niej).

■ LEFT (left[most substring] — lewa część łańcucha) ***

FUNKCJA

LEFT (a\$,l [,b\$])

Parametrami są tu: a\$ — łańcuch, b\$ — łańcuch 1-znakowy, l — liczba całkowita. Funkcja LEAVE tworzy łańcuch złożony z l pierwszych znaków łańcucha a\$. Jeśli długość łańcucha jest mniejsza niż podana liczba l, wówczas na końcu łańcucha wynikowego dopisywane są spacje. Jeżeli użyjesz parametru b\$ — w miejsce spacji zostaną dopisane znaki zawarte w łańcuchu b\$. Przykład:

SAY LEFT('1234567', 4)

(da w wyniku łańcuch '1234'), natomiast

SAY LEFT('123456,9', '')

(wydrukuje łańcuch '123456***').

■ LENGTH (length [of string] — długość łańcucha) ***

FUNKCJA

LENGTH(a\$)

gdzie: a\$ — to łańcuch.

Oblicza długość łańcucha a\$, wliczając w to ewentualnie występujące w nim spacje. Przykładowo:

SAY LENGTH ('Commodore') (da w wyniku 9), zaś

SAY LENGTH ('Amiga 500')

(da w wyniku również 9 [gdyż spacja zostanie wliczona]).

■ **LINES** ([number of] lines — liczba wierszy) *** FUNKCJA
LINES(a\$)

gdzie: a\$ — łańcuch.

Podaje liczbę wierszy wpisanych uprzednio do zbioru, który jest aktualnie przyporządkowany polu interaktywnemu. Przykład:

PUSH 'to jest wiersz'

PUSH 'a to następny'

SAY **LINES**(stdin)

(da w wyniku 2).

■ **MAX** (max[imum]) *** FUNKCJA

MAX (n1,n2 [,n3, ... , nn])

Znaczenie parametrów: n1 — nn cyfry.

Znajduje wartość maksymalną spośród dwóch lub większej liczby cyfr, które są argumentami funkcji. Przykład:

SAY **MAX**(2,7,-4,12,8,15,3,9)

(da w wyniku 15).

■ **MIN** (min[imum]) *** FUNKCJA

MIN (n1,n2 [,n3, ... , nn])

gdzie: n1 ... nn — cyfry.

Analogicznie jak **MAX** — znajduje najmniejszy spośród podanych argumentów cyfrowych. Przykład:

SAY **MIN** (4,8,-2,7,18,0,-8,12,4)

(da w wyniku -8).

■ **NOP** (n[o] op[eration] — nie rób nic) *** INSTRUKCJA

NOP

Ta instrukcja nie robi nic. Jest jednak konieczna, gdy chcesz użyć "pustej" instrukcji **ELSE** np. w zagnieżdżonych instrukcjach **IF ... ELSE** (użycie samego **ELSE** spowodowałoby błąd polegający na powrocie do pierwszej instrukcji **IF**. Przykład: (przy próbie nie wpisuj komentarzy w nawiasach, które są konieczne dla zrozumienia działania przykładu):

IF i=j **THEN**

(to jest zewnętrzna instrukcja **IF**)

IF j=k **THEN** a=0

(to jest wewnętrzna instrukcja **IF**)

ELSE NOP

(ta instrukcja **ELSE** związana jest z wewnętrznym **IF**)

ELSE a=a+1

(natomiast to **ELSE** związane jest z zewnętrznym **IF**).

■ **NUMERIC** ([set] numeric [precision] — ustaw precyzję) *** INSTRUKCJA

składnia 1: **NUMERIC** <**DIGITS** | **FUZZ**> w1

(gdzie: w1 — wyrażenie),

składnia 2: **NUMERIC FORM** <**SCIENTIFIC** | **ENGINEERING**>

Instrukcja (w zależności od użytej składni i słowa kluczowego) ustawia format i precyzję dla stosowanych w dalszej części programu wyrażeń numerycznych. Ustalony w ten sposób format i precyzja będą obowiązywać również przy przekazywaniu wartości jako argumentów funkcji wewnętrznych. Przy składni pierwszej:

* **DIGITS** w1 — ustawia precyzję (liczbę miejsc, jaka będzie obliczana po przecinku w obliczeniach arytmetycznych na wartość obliczoną przez wyrażenie w1. UWAGA: Zamiast wyrażenia możesz podać wprost liczbę. Wartość w1 musi być całkowitą liczbą dodatnią. Przykład:

NUMERIC DIGITS 14

(obliczenia arytmetyczne będą prowadzone do 14 miejsca po przecinku),

* **FUZZ** w1 — liczba miejsc dziesiętnych oznaczona przez wyrażenie w1 będzie ignorowana przez program podczas obliczeń, np.:

NUMERIC FUZZ 4

a=1.23456 ; b= 4.59824

SAY a*b

(da w wyniku 5.4, a nie jak można by się spodziewać 5.67680318294).

Dla składni 2:

* **NUMERIC FORM SCIENTIFIC** — określa, że wszystkie wyniki od tej pory będą podawane w notacji naukowej, zap:

* **NUMERIC FORM ENGINEERING** — w notacji inżynierskiej.

[Notacja naukowa to przedstawienie liczby jako mantysy pomnożonej przez 10 do jakiejś potęgi, np. 1.23E3 to jest 1230 w tej notacji. W notacji inżynierskiej mantysa jest z zakresu 1-1000, zaś wykładnik jest wielokrotnością liczby 3, np 71D3 to 357911]. Zakładając że a=12345 otrzymamy: przy

NUMERIC FORM SCIENTIFIC

SAY a

(da w wyniku 1.2345E4), zap przy

NUMERIC FORM ENGINEERING

SAY a

(da w wyniku 23.111D3).

■ **OPEN** (otwórz) *** FUNKCJA

OPEN(a\$,b\$ [, 'A' | 'R' | 'W']

gdzie:a\$, b\$ — łańcuchy.

Otwiera zbiór zewnętrzny, aby wykonać na nim oznaczone działanie. a\$ — oznacza nazwę, pod jaką dany zbiór będzie wywoływany w dalszej części programu, b\$ — jest nazwą wtórną (zawiera np. nazwę, pod jaką zbiór występuje na dysku, urządzenie, na którym jest zainstalowany, itp.). UWAGA: Jak pokazuje przykład, możesz otwierać przez **OPEN** nie tylko zbiory dyskowe. Nie ma ograniczenia co do liczby otwieranych przez **OPEN** zbiorów zewnętrznych w jednym programie. Wszystkie otwarte zbiory zostają automatycznie zamknięte w momencie wyjścia z programu. Funkcja przyjmuje wartość "true" (1), gdy operacja otwarcia się powiedzie, lub "false" (0), gdy wystąpi błąd. Słowa kluczowe oznaczają:

* **A** — otwarcie zbioru na dopisywanie danych,

* **R** — otwarcie zbioru na odczyt,

* **W** — otwarcie zbioru na zapis. Oczywiście jeśli otwieramy urządzenie inne niż zbiór dyskowy — wówczas opuszczamy słowo kluczowe. Przykład:

SAY **OPEN**('MojeOkno','CON:160/50/320/100/MojeOkno/cds')

(otwiera okno o podanych parametrach [opis tych parametrów znajdziesz przy opisie instrukcji **WINDOW** Amiga **BASIC**/ tom II tej książki] i przyjmuje wartość 1, jeśli okno zostanie otwarte),

SAY **OPEN**('wyniki','df1:roboczy','W')

(otwiera na dysku df1: zbiór o nazwie "roboczy" przeznaczony na zapis. Jeśli zbiór istnieje i nie jest zabezpieczony przed zapisem — wówczas funkcja przyjmie wartość 1, w innym przypadku 0. Od tej chwili wszystkie odwołania do zbioru "wyniki" spowodują zapisywanie danych w zbiorze o nazwie "roboczy", chyba że zamkniesz go przez **CLOSE**).

■ **OPTIONS** (opcje, inne możliwości) *** INSTRUKCJA

OPTIONS [**NO**] [**FAILAT** w1 | **PROMPT** w2 | **RESULTS** | **CACHE**]

gdzie: w1 — wyrażenie.

Pozwala na ustawienie wartości początkowych w programie. Jeśli nie użyjesz żadnego ze słów kluczowych — wszystkie opcje kontrolowane przez słowa kluczowe zawarte w instrukcji przyjmą wartości wstępnie ustawione (default). Poszczególne słowa kluczowe (pamiętaj, że jeśli użyjesz jednego z nich, wówczas nie możesz już w tej samej instrukcji użyć drugiego) oznaczają: * **FAILAT**:w1 — liczba całkowita w1 określa poziom, który spowoduje traktowanie kodu zwrotnego rozkazu jako błędu (podobnie jak w **ADOS**),

* **PROMPT** w2 — tu wyrażenie w2 jest łańcuchem, który będzie zastosowany jako wskaźnik gotowości systemu (prompt) w instrukcjach **PULL** lub **PARSE PULL**,

* **RESULTS** — "wskaże" interpreterowi na konieczność użycia łańcucha wynikowego przy przekazywaniu rozkazów do programu zarządzającego,

* **CACHE** — uaktywnia wewnętrzną procedurę cache'owania (UWAGA: włączona jest ona defaultowo przy starcie programu i przyspiesza pracę interpretera. Przed każdym z tych słów kluczowych możesz użyć słowa **NO**, które spowoduje przywrócenie defaultowej wartości dla danej opcji. Przykład:

* **OPTIONS FAILAT** 10 (jepli kod zwrotny instrukcji wynosi 10 lub więcej — wystąpi komunikat o błędzie), zaś

OPTIONS PROMPT "Słucham Cię mój Władco?" (spowoduje, że jeśli będzie potrzebna Twoja interwencja, komputer w tak miły dla oka sposób poprosi o łaskawe dotknięcie klawiatury).

Marek Pampuch

Każdy czasami ma potrzebę podzielenia się z bliźnimi swoimi uwagami lub genialnymi pomysłami czy wyżalenia się na cały świat. Najczęściej jednak w takiej chwili miotamy się bezsilnie przed lustrem. Nikt nie chce nas słuchać, zaś redakcje grzecznie informują, że "nadesłany tekst nie nadaje się". Bezsilna złość może czasem rodzić agresję. Pragnąc dać upust temu, co w każdym z nas siedzi — zapraszamy do Amigowego Hyde Parku. Będziecie w nim mogli wypowiadać się ze wszy-



Hyde Park

stkiego, co Wam leży na sercu. Aby jednak Wasze teksty mogły być zamieszczone w tej rubryce, muszą spełniać cztery warunki:

- muszą być w jakikolwiek sposób związane z Amigą,
- nie mogą być dłuższe niż jedna strona,
- aby Hyde Park nie kojarzył się z o-hyde-ą — tekst nie może zawierać słów powszechnie uznanych za niecenzuralne,
- list nie może być anonimowy.

Po przeczytaniu tego ostatniego warunku nieco odeszła Wam ochota na wykrzyżowanie się. Aby jednak umożliwić Wam pełną swobodę wypowiedzi, zamieszczane teksty nie będą podpisywane (adres pozostaje do wiadomości redakcji). Jest jednak jeden wyjątek od tej reguły. Jeżeli tekst będzie zawierał spersonifikowany atak, wówczas zostanie on opatrzone nazwiskiem autora, aby "ofiara" napaści miała prawo obronić

się (na tych samych łamach). Nie będziemy także nadesłanych tekstów opatrywać komentarzem (choć czasami aż język świerzbi).

Teksty należy nadsyłać na adres redakcji w kopertach z dopiskiem HYDE PARK. UWAGA: REDAKCJA NIE PONOSI ŻADNEJ ODPOWIEDZIALNOŚCI ZA ZAMIESZCZONE W TEJ RUBRYCE TEKSTY.

Na zakończenie — pierwsze przykładowe teksty z dziedziny Amigowskiej demokra-

cji (nie podpisane i podpisy-ny):

* *Czy to prawda, że Commodore zaprzestał produkcji Amigi 500? Chyba zgłupieli. Zróbcie coś z tym fantem, o ile możecie. Proponuję założyć "Klub ratowania pięćsetki". Można by urządzić pikietowanie Commodore lub strajk głodowy, albo zaskarżyć tę decyzję do Rady Europy. My, Polacy, mamy w takich sprawach doświadczenie.*

* *Zamieszczajcie więcej opisów programów użytkowych. Niebiescy (tzn. IBM-owcy) do każdego programu mogą kupić sobie książkę, a my Amigowcy to od macochy? Wiem, że nie*

opłaca się wydanie książek w nakładzie 10.000 egzemplarzy, ale wasze pismo mogłoby coś w tej dziedzinie zrobić.

* *Ja już nie mogę. Mój starszy brat Jurek jest paskudny. Jak ja chcę sobie pograć, to mówi, że Amiga nie jest dla smarkaczy i chowa dżojstik. Konradek Wytrzysszczyński, Lipowo Śródleśne*

Tyle w sprawie Hyde Parku ma do powiedzenia redakcja. Przyszłość tej rubryki zależy tylko od Was. Czekamy na listy.

PZEGLĄD PROGRAMÓW KOPIUJĄCYCH

(CZ. 2)

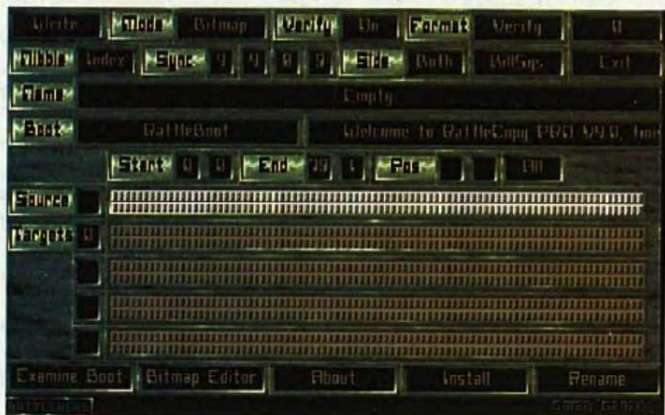
W poprzednim odcinku poznaliście zasady zapisu na dysk i niektóre programy kopiujące, dziś omówimy pozostałe, a także pokazemy Wam, jak wyglądają na ekranie Amigi. Dowiedzie się także, w jakim czasie dyskietki są kopiowane przez te programy. Lektura obu części artykułu pozwoli Wam na wybór najbardziej odpowiedniego programu. W tym numerze kończymy artykuł — nie oznacza to jednak, że do tematu nie powrócimy (najprawdopodobniej z nieco bardziej profesjonalnym podejściem).

Thomas Lopatic

RATTLE COPY

Ten program kopiujący jest rozprowadzany jako Shareware. Jest to dosyć szybki,

porównywalny z Xcopy, kopier. W rattle copy posiadacz (minimum 1 MB) może wczytać program do RAM-u, a stamtąd przesać go na dowolną liczbę dyskietek. Funkcja RAM-Copy tego programu jest szybsza niż analogiczna funkcja Xcopy.



Podobnie jak w przypadku XCopy, można zamiast całej dyskietki kopiować tylko wybrane ścieżki. Program ma opcję HELP służącą jako ściągą dla zapominalskich.

PCOPY

Użytkownicy, których nie stać na zakup programów kopiujących, muszą się zadowolić kopierami z rodzaju Public Domain. Znajdują się wśród nich całkiem niezłe programy kopiujące. Alternatywą dla przedpotopowego rozkazu DiskCopy może być program o nazwie Pcopy. Różni się on od omawianego rozkazu między innymi tym, że PCOPY można obsługiwać za pomocą myszki.

TURBO BACKUP

Kolejny program kopiujący rodzaju Public Domain nazywa się Turbo backup. Także i ten program swoją szybkością i funkcjonalnością wielokrotnie przewyższa rozkaz DiskCopy. Może pracować z dwiema do czterech stacji dysków. Nie ma tu jednak, znanej Ci choćby z XCopy możliwości jednoczesnego zapisu na kilku dyskietkach, a ponadto, ku zmartwieniu większości użytkowników Amigi, program ten nie będzie działał z jedną stacją. W przypadku napotkania na błąd przy odczycie lub zapisie — kopiowanie zostaje

przerwane, jednak używając opcji RETRY możesz spróbować naprawić błąd. Nie zrażaj się, jeśli błąd się powtórzy. Trafiła mi się kiedyś dyskietka z błędem, na którym "wylatywały" wszystkie programy kopiujące (Turbo backup też). Ponieważ autor programu twierdził, że to ostatnie jest niemożliwe, a mnie zależało na programie — wykonałem test "na wytrzymałość". I rzeczywiście, błąd został naprawiony! Wprowadziłem dopiero po dwudziestym drugim (to nie pomyłka!) kliknięciu opcji RETRY, ale jednak. Tak uzyskana kopia służy mi już dosyć długo. Turbo backup teoretycznie nie powinien kopiować programów zabezpieczonych, jednak zdarzają się wyjątki potwierdzające regułę. W naszym teście był jeden taki przypadek. Program działa nieco wolniej niż opisywane wyżej kopierki (na przykład Rattle Copy czy XCopy), a to dlatego, że jako jedyny nie uznaje kopiowania bez weryfikacji. Każda próba wyłączenia jej powoduje wyświetlenie komentarza, że kopiowanie bez weryfikacji jest tylko stratą czasu, i ... w takim przypadku nic nie zostanie skopiowane. Przeprowadzone przeze mnie testy wykazują, że jest to jeden z najpewniejszych (jeżeli nie najpewniejszy) programów kopiujących.

WHITE LIGHTNING

Jeden ze starszych programów kopiujących. Bardzo łatwy w obsłudze, a przy tym bardzo pewny. Jeśli już coś uda się nim skopiować, wówczas możesz być pewien, że tak uzyskana kopia będzie Ci długo służyć. Nieco słabszą stroną programu jest formatowanie. Wprowadzi funkcja FAST FORMAT wykonuje swoje zadanie bardzo szybko, jednak dyskietka nie zawsze zostanie sformatowana właściwie. Ciekawą cechą White lightning jest to, że nie kopiuje on całej dyskietki, a jedynie te sektory, które zawierają jakieś dane. Dzięki temu użytkownik oszczędza czas. Z drugiej strony program nie pozwala na wykrzestanie dodatkowych stacji dysków. Jeśli już miałbym go



komuś polecać, to jedynie użytkownikom Amig bez rozszerzenia pamięci.

BURSTNIBBLER 1.0

To chyba najszybszy program kopiujący... pod warunkiem, że kopiujesz w trybie DOS copy. Zastosowane tu procedury stały się podstawą programu wykorzystywanego w SYNCRO EXPRESS, stąd, jak łatwo się domyślić — szybkość kopiowania niewiele ustępuje temu ostatniemu, mimo że Burstnibbler nie korzysta z żadnego dodatkowego sprzętu. Ale nie ma róży bez kolców. Kopiowanie w tym trybie, choć szybkie — nie jest weryfikowane, co praktycznie pozwala na kopiowanie tylko nie zabezpieczonych dyskietek DOS-owskich. Jakby dla równowagi, drugi tryb kopiowania (Deep copy) ma za równo weryfikację, jak i bardzo głęboką analizę zabezpieczeń. W momencie pojawienia się programu nie było na rynku zabezpieczenia, które mogłoby się Burstnibblerowi oprzeć, zwłaszcza, że ma on możliwość kopiowania zarówno do 81 ścieżki, jak i (do pewnego stopnia) przedłużonych ścieżek. Za tę wygodę trzeba jednak płacić. Aby uzyskać kopię dyskietki w trybie Deep copy, program potrzebuje aż 274 sekund. Dodatkowo, program jest na tyle "nieinteligentny", że nawet wówczas, gdy pomyłkowo będziesz usiłował skopiować nie zabezpieczoną dyskietkę w tym trybie, to i tak będzie ona "analizowana".

Oprócz wersji PD (1.0) istnieją także nieco ulepszone wersje, które w założeniu miały być komercyjne. Próby wprowadzenia ich na rynek spowodowały jednak liczne protesty producentów oprogramowania, stąd od dłuższego czasu nie pojawiła się żadna jego nowa wersja.

A COPIER

Pierwszy kopier na Amigę z rodzaju Public Domain. Stary i niezbyt dobry. Kopiowanie A copier-em jest niewiele szybsze od kopiowania na poziomie Amiga DOS-u, nie ma weryfikacji, zaś fakt, że dane są odczytywane z dysku dwukrotnie wolniej niż zapisywane na kopii, nie wróży długiego życia kopii tak uzyskanej. Ponadto — program wczytuje do pamięci jedynie 23 ścieżki i pracuje tylko z wbudowaną stacją df0: a zatem kopiowanie wiąże się z koniecznością czterokrotnej wymiany dyskietki. W okresie, gdy na rynku pojawiają się coraz nowsze i lepsze programy kopiujące, A COPIER nie powinien zastępować nawet na krótką wzmiankę, jednak uznałem, że w przeglądzie kopierów powinny się znaleźć nie tylko "brylanty".

Mówiąc o programach kopiujących, nie można pominąć tych, które pozwalają na uzyskanie kopii bezpieczeństwa pojedynczych zbiorów. Niestety, z powodu niewielkiej ilości miejsca przeznaczonego na ten artykuł nie można ich szerzej omówić. Mimo, że istnieje kilka specjalizowanych programów tego rodzaju, jednak standardem w zakresie kopiowania plików stał

się program, dla którego takie kopiowanie jest jedynie drobną częścią jego funkcji. Jak już się pewnie domyślicie, chodzi oczywiście o Disk master. Być może omówię ten program dokładnie (bo warto!) w którymś z kolejnych numerów Magazynu AMIGA, zwłaszcza że pojawiła się jego wersja 2.0 — zawierająca wiele ulepszeń w stosunku do najbardziej rozpowszechnionej wersji 1.3. UWAGA: Spotykana gdzieś tam wersja 3.0 — chociaż wykonuje podobne operacje — jest jednak zupełnie innym programem, napisanym przez innych ludzi i rozprawdzanym na zasadach niekomercyjnych. Program pozwala na kopiowanie bez weryfikacji zbiorów znajdujących się na dyskietkach o formacie DOS-owskim. Próba skopiowania dyskietek zabezpieczonych w najlepszym przypadku spowoduje komunikat: Bad Directory. Aby skopiować zbiór, wystarczy w jednym oknie katalogowym ustawić dysk "źródłowy", w drugim "docelowy" (może to być także RAM-disk, co jest wygodne w przypadku posiadania jednej stacji dysków), zaznaczyć za pomocą myszki programy, które chcemy skopiować i wybrać z repertuaru rozkaz COPY lub MOVE. W tym ostatnim przypadku zbiór po skopiowaniu zostanie usunięty z dysku "źródłowego". Można także kopiować zawartość całego podkatalogu — zaznaczając jego nazwę. W razie potrzeby można też sformatować dyskietkę w dowolnej stacji. Jak już wspominałem, Disk master ma wiele innych funkcji, jednak nie one są tematem tego artykułu.

DIRECTORY OPUS

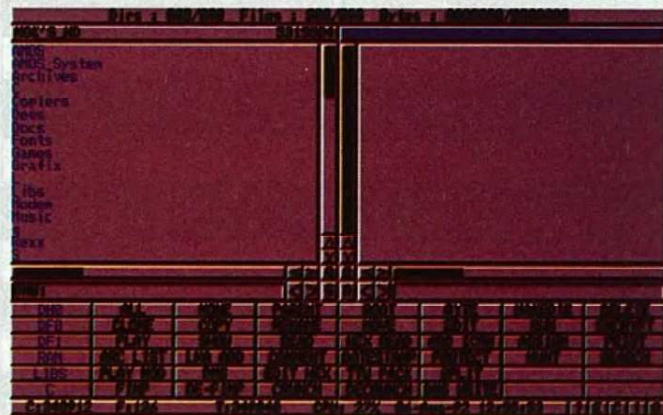
Program ten — podobny w działaniu do Disk Mastera pojawił się na rynku stosunkowo niedawno i (chyba trochę na wyrost) został okrzyknięty następcą tamtego programu. Czy różni się Directory Opus od Disk Mastera w interesującym nas zakresie kopiowania zbiorów i formatowania dyskietek? O tym także przeczytacie w zapowiadającym artykule.

CLIMATE

Na podobnych zasadach, jak oba opisane programy, działa starszy program tego samego typu Climate. Nie jest on jednak tak chętnie stosowany, gdyż ma kilka cech powodujących mniej komfortową obsługę, niż w przypadku Disk Mastera. Jest to między innymi denerwujący chyba wszystkich użytkowników tego programu "fastdir" — czyli zbiór zawierający uporządkowany katalog, który zapisuje się na dysku. Próba kopiowania Climate -m zbiorów z katalogu, w którym jest więcej niż 144 pliki, zakończy się (ze względu na taką właśnie pojemność okna katalogowego) fiaskiem. Przy kopiowaniu całych katalogów należy o kilka razy więcej niż w Disk-Masterze potwierdzić ten zamiar (za pomocą myszki).

Istnieją także mniej chętnie wykorzystywane programy przeznaczone do kopiowania, jak Dir Util, Amigutil czy File copy.

Do "rodziny" programów kopiujących zaliczają się także programy pozwalające na wykonanie tzw. backupu twardego dysku. Mimo



SCH.



iz od czasu niezbyt milej przygody z wirusem XENO, zakończonej sformatowaniem twardego dysku, bardzo chętnie używam takich programów, jednak pominałem je w niniejszym zestawieniu, bo chociaż również służą do uzyskania kopii bezpieczeństwa, to jednak ze względu na sposób ich działania — nie mogłem ich poddać testom polegającym na wykonywaniu kopii dyskietek. Uprzedzając słusne protesty posiadaczy "twardzieli", mogę im tylko przyrzec, że Magazyn AMIGA o nich nie zapomni.

UWAGA: Podczas testowania programów kopiujących używaliśmy zestawu następujących oryginalnych (zabezpieczonych i nie zabezpie-

czonych przed kopiowaniem) programów:

Workbench 1.3D, Becker-text, HED, SIDMon 2, MusicX, Dragons Lair, Dungeon Quest, Flipit, Diskmaster (wersja PAL), International Soccer, Grand Prix Circuit, Kings Quest IV, Dr. Plummer House of Izux, Romance of three Kingdoms, Crack.

W poniższej tabelce podany jest średni czas kopiowania (bez czasu zużytego na zamianę dyskietek). Jeśli czas ten nie jest podany, oznacza to, że program nie może wykonać kopiowania w danym trybie.

Amiga Magazyn 7/1991
Tłum. Marek Pampuch

Nazwa programu kopiującego	Czas kopiowania 1 dyskietki bez weryfikacji (tryb DOS)	Czas kopiowania 1 dyskietki z weryfikacją (tryb DOS)	Czas kopiowania 1 dyskietki w trybie DEEP lub NIBBLE
Synchro Express	56 s	---	157 s
Burst Nibbler	61 s	---	274 s
Tetra copy	68 s	106 s	---
PCopy	69 s	99 s	---
Rattle Copy	70 s	105 s	---
XCOPY	70 s	108 s	---
White Lightning	72 s	117 s	---
Cyclone	73 s	---	189 s
Rozkaz DISCOPY	79 s	146 s	---
Project D	91 s	123 s	209 s
Black Copy	106 s	---	---
Turbo Backup	---	103 s	---
ACopy	110 s	---	---

Giełda AMIGI Regulamin

- Ogłoszenie w Giełdzie AMIGI może zamieścić każdy.
- Ogłoszenie jest płatne przed opublikowaniem — do listu z treścią należy dołączyć kopię potwierdzenia wpłaty 50.000 zł na konto: LUPUS Sp. z o.o. PKO BP IX O/Warszawa, r-k 1599-318121-136.
- Od opłaty zwolnieni są prenumeratorzy, którzy przy treści ogłoszenia powinni podać numer prenumeraty.
- Ogłoszenia ukazują się w trzech rubrykach: SPRZEDAM, KUPIĘ, NAWIĄŻĘ KONTAKT.
- Ogłoszenia SPRZEDAM muszą zawierać cenę i nie mogą to być ogłoszenia dotyczące działalności gospodarczej kogoś, kto produkuje dany artykuł czy pośredniczy w handlu nim.
- Ogłoszenia SPRZEDAM niezgodne z wymienionymi ograniczeniami nie będą zamieszczone, a wpłacone za nie pieniądze przepadną.
- Ogłoszenia NAWIĄŻĘ KONTAKT powinny zawierać wyszczególniony obszar zainteresowań.
- Redakcja nie ponosi odpowiedzialności za treść ogłoszeń ani skutki z nich wynikające.
- Na kopercie prosimy umieszczać dopisek "Giełda AMIGI". Wszelkie sprawy można także załatwić w siedzibie redakcji.

SPRZEDAM

Amigę 500 (1MB) gwarancja, monitor, literaturę, dyskietki itp. — 10 mln zł
K. Loster, ul. Pocztowa 4a, 34-410 Rabka, tel. 76864 (po południu). [pa1]

Samplery:
Stereo (mono 32 kHz, stereo 16 kHz) — 300 tys. zł
Mono (17 kHz) — 195 tys. zł
Oraz MIDI (2xOUT, 1xIN) — 200 tys. zł
Programator EPROM-ów do Amigi — 1.5 mln zł
Marcin Skoneczko, ul. Rąbieńska 24, 94-227 Łódź. [pa2]

Mówiący mikrokomputer, mówiący gong, mówiący notatnik lub zegar, budzik. Współpracuje

z każdym komputerem lub samodzielnie i jako samodzielny sampler. Wystarczy dołączyć RAM i wgrać tekst przez mikrofon. Wszelkie w jednym układzie scalonym. Układ+plytka+opis — 240 tys. zł

Jerzy Andreasik, ul. Spółdzielców 10/3, 57-320 Polanica Zdrój. [pa3]

Amiga 500 v. 1.3 z rozszerzeniem pamięci do 1 MB, monitor monochromatyczny firmy Philips, dyskietki, dodatkowa literatura, mouse-pad, joystick, kabel Amiga-Euro — 5.950.000 zł
Sebastian, Katowice 588-674.

[pa4]



Norbert Spittenandt

W pierwszym odcinku zajmowaliśmy się głównie instrukcjami PSET i LINE. Teraz pójdziemy nieco dalej. Poznamy kolejne rozkazy graficzne Amiga BASIC-u (na przykład CIRCLE), a ponadto postaramy się, aby nasze rysunki wykorzystywały możliwości Amigi w zakresie barw. Aby ułatwić zrozumienie części teoretycznej, przygotowaliśmy, podobnie jak w pierwszej części kursu, kilka krótkich programów. Ponownie sprawdzicie, dzięki prostym ćwiczeniom, zasób nabytej wiedzy.

Co to jest kolor i w jaki sposób widzi go nasze oko? Światło jest promieniowaniem. Z fizycznego punktu widzenia takim samym jak na przykład stosowana w kuchenkach mikrofała lub promieniowanie Roentgena. Różni się jednak od wymienionych promieniowań długością fali. Światło widziane przez ludzi ma długość fali od 380 do 720 nm. Promieniowanie to pada na siatkówkę oka, którą można sobie wyobrazić jako raster składający się z czopków. Każdy z czopków ma w sobie pigment barwny i przez to różnie reaguje na fale różnej długości. Inne impulsy nerwowe wytworzy czopek, na który padnie promieniowanie o większej długości fali, a inne — przy promieniowaniu "krótszym". Sygnały te są przetwarzane przez "centrum wzroku" naszego mózgu i w efekcie nałożenia się na siebie tysięcy impulsów otrzymujemy obraz. Na "krajcach widzialności" leżą kolory — z jednej strony fioletowy, a z drugiej zielony i pomarańczowy. Właściwie to istnieją tylko trzy kolory podstawowe. Co do czerwonego i niebieskiego panuje jednoznaczność, natomiast różne szkoły wiodą spór, czy tym trzecim ma być żółty, czy zielony. W technice wizyjnej przyjęto, że jednak ten ostatni. Pozostałe kolory otrzymujemy przez odpowiednie mieszanie barw podstawowych.

Istnieją dwie metody mieszania barw. Pierwsza z nich — addytywna (od ang. add —

Programowanie grafiki

(cz. 2)

KOLOROWY PITAGORAS

Powiadają, że świat jest kolorowy. Dlaczego? Przecież wszystko tworzą bezbarwna materia i taka sama energia. Ale przecież kolor jest, a jaki jest, każdy widzi. W kolejnym odcinku naszego kursu pokażemy Wam, jak w BASIC-u można programować kolorowe obrazki.

dodawać) stosowana jest na przykład w telewizji kolorowej, wymaga jednak kolorów podstawowych w standardzie RGB (ang: red, green, blue — czerwony, zielony, niebieski). Metoda subtraktywna działa na innych zasadach. Jako koloru podstawowego używa barwy białej. Światło, przechodząc przez filtr danego koloru, traci składnik tej samej barwy.

Wystarczy tej teorii. Spróbujmy zobaczyć, jak ma się ona do tematyki naszego kursu. Przy uruchomieniu BASIC "widzi" tylko 4 kolory przejęte z Workbencha: czarny, biały, pomarańczowy i niebieski. Są to "aż" 4 barwy (numerowane od 0-3). Spróbujemy zmienić te kolory. Umożliwia to instrukcja PALETTE. Potrzebuje ona czterech parametrów: numeru zmienianego koloru oraz trzech parametrów określających intensywność trzech podstawowych barw w kolorze, jaki chcemy uzyskać. Jak już wiemy, na razie pierwszy parametr może się zmieniać w zakresie 0-3, natomiast trzy następne muszą być liczbą z zakresu od 0 do 1 (przy ustawianiu tych parametrów przyjmuje się skok = 1/15, co umożliwia nam uzyskanie 16 wartości). Wartość = 1 oznacza pełną intensywność danego składnika, 0 — brak

składnika w tworzonym kolorze. Nietrudno się chyba domyślić, że parametry dotyczą kolejno: najpierw składnika czerwonego, potem zielonego, zaś ostatni z nich — niebieskiego. Jeśli w PALETTE określimy, że wszystkie składniki RGB mają tę samą intensywność, na przykład:

```
PALETTE 2,.5,.5,.5
```

wówczas w wyniku powstanie kolor szary o średniej jasności.

```
PALETTE 2,1,.5,.5 da w wyniku
kolor jasnoczerwony, zaś PALETTE
2,1,0,1 — czysty fiolet.
```

Wprowadzanie parametrów określających zawartość poszczególnych składników RGB w zakresie od 0 do 1 jest uciążliwe. Wpiszcie programik Color Setter. Umieszczona w nim procedura SETRGB pozwoli na podawanie tych parametrów w zakresie 0-15. Na przykład:

```
SETRGB 2,8,8,8
```

pozwoli na utworzenie takiego samego szarego koloru, jak opisana wyżej instrukcja PALETTE 2,.5,.5,.5

Program Color Setter został pomyślany jako pomoc naukowa dla eksperymenta-

torów. Pyta on o trzy składniki barw i na tej podstawie rysuje prostokąt z odpowiednią mieszanką kolorów. W sumie możliwe jest 16 x 16 czyli 4096 różnych kombinacji. Pobawmy się. Ustawcie wartość parametru określającego zawartość jednego lub dwóch składników barwy i obniżajcie pozostałe stopniowo o 1; albo — ustawcie składniki barw na 0 i podwyższajcie stopniowo o 1, a potem ustawcie jakieś składniki barw na dowolną wartość i zmieniajcie stopniowo pozostałe. Podprogram GETRGB w listingu Color Setter działa w sposób przeciwny niż opisany SETRGB. Podaje on wartości parametrów określających składniki RGB dla aktualnego koloru. Porównajmy wyniki programu z niewielką liczbą kolorów, jakie standardowo oferuje BASIC.

* W modelu RGB kolor powstaje przez addytywne mieszanie różnych odcieni kolorów podstawowych: czerwonego, zielonego i niebieskiego. Model ten jest modelem "technicznym" nie przypominającym w żaden sposób metod, jakich używamy na przykład przy malowaniu obrazów, kiedy bierzemy jakiś kolor i mieszamy go w odpowiednim stosunku z białym lub czarnym. Na tym sposobie pracy oparty jest drugi stosowany model o nazwie HSV. Poszczególne składniki nazwy tego modelu oznaczają:

H — to odcień (ang. hue) — jest określany jako kąt z zakresu od 0 do 359 stopni, który pokrywa spektrum barw uporządkowane w formie koła. Kąt równy 0 stopni oznacza kolor na lewym końcu spektrum (pomarańczowo-czerwony), 180 stopni w środku (niebieski), a 360 stopni oznacza kolor z prawego końca spektrum barwnego (czerwono-fioletowy).

S — to nasycenie (ang. saturation). Barwy mają niekolorowy (biały, czarny lub szary) i kolorowy składnik. Przykładowo: kolor ustawiony w naszym programie parametrami SETRGB 2,10,8,8 ma składnik szary o wartości 8, zaś składnik czerwony równy 2. Parametr S w modelu HSV ma wartość z zakresu 0 do 1 oznaczający zawartość składnika kolorowego. Wartość 1 oznacza maksymalny stopień nasycenia (brak składnika

```

WHILE e$ <> "n"
  INPUT "Rot-Komponente: "; r%
  INPUT "Grün-Komponente: "; g%
  INPUT "Blau-Komponente: "; b%
  SETRGB 3, r%, g%, b%
  LINE (50,50)-(350,150),3,bf
  LOCATE 21,1
  INPUT "Noch ein Versuch? Ende="
  <n> "; e$
  CLS
WEND
END

```

```

SUB SETRGB(nr%, r%, g%, b%) STATIC
  PALETTE nr%, r%/16, g%/16, b%/16
END SUB

```

```

SUB GETRGB(nr%, r%, g%, b%) STATIC
  IF nr% > WINDOW(6) THEN ERROR
  7
  ct% = PEEKL(PEEKL(PEEKL(WINDOW(
  7)+46)+48)+4)
  f% = PEEKW(ct%+2*nr%)
  r% = (f% AND 3840)/256
  g% = (f% AND 240)/16
  b% = f% AND 15
END SUB

```

COLORSETTER ko-
lory komponowane ze skład-
ników RGB.

białego). Przykładem wysokiego nasycenia jest często stosowany w malarstwie błękit pruski. Relatywnie niskie nasycenie (i dużą zawartość bieli) ma natomiast kolor jasnoniebieski. Najwyższe nasycenie ma 8 barw podstawowych i wszystkie inne, w których składnik barwny jest zerowy (na przykład czarny). Zmniejszając parametr S, powiększamy jednocześnie zawartość bieli w kolorze.

V (ang. value) jest wartością oznaczającą jasność koloru. 0 jest najmniejszą jasnością, czyli po prostu czernią. Przy najwyższej wartości V (1) otrzymujemy biel (1). Gdy zmniejszamy ten parametr, nasz kolor uzyskuje coraz większą domieszkę czerni.

Aby pokazać, w jakim stopniu jest to praktyczne, w programie Set Circle umieściliśmy podprogram: SETHSV. Zastosowaliśmy go w celu utworzenia barwnego kółka. Aby zamienić wartości z modelu HSV na zrozumiałe przez komputer, odpowiadające im, wartości modelu RGB, dołączone są podprogramy RGBTOHSV i HSVTORGB. W ten sposób można zamienić model HSV na model RGB

Składnia rozkazu CIRCLE

CIRCLE (x,y), promień, kolor, kąt początkowy, kąt końcowy, współczynnik proporcji obrazu.

Teraz przejdziemy do następnego tematu. "Nie niszczyć moich kół" — rzekł grecki uczoney Archimedes do rzymskich żołnierzy, którzy zdeptali mu jego rysunki na piasku. Legenda mówi, że wielki matematyk nie wyszedł dobrze na zrobieniu tej uwagi. Naszego CIRCLE nikt nie niszczy. Instrukcja BASIC-a — CIRCLE — rysuje jednak... elipsę, bowiem okrąg jest specjalnym rodzajem elipsy. Spróbujmy to zrobić. Na początek ustawmy na ekranie takie same okna LIST i wynikowe, jak to zrobiliśmy w poprzednim odcinku naszego kursu. Teraz wpiszcie w oknie LIST:

```
CIRCLE (314,93),90
```

i uruchomcie program naciskając jednocześnie klawisze [prawy Amiga] i [R]. Nasz "program" narysuje koło mniej więcej na środku okna, o promieniu równym 90 pikseli. "No dobrze, ale takie okrągłe to koło jednak nie jest" — przyczepi się nasz dociekliwy Czytelnik. Czyżby zależało to od ustawienia ekranu? Cierpliwości. Zobaczymy to za chwilę. Na razie mamy kolejny problem. Dlaczego nasze "koło" musi mieć promień 90

pikseli? Na to pytanie odpowiedź jest prosta. Ponieważ nasze okno wynikowe ma wysokość 187 punktów ekranowych, staraliśmy się dopasować jego wielkość do tej wysokości. Ale to jakoś nie pasuje. Oto dowód:

Wpiszcie pod powyższą instrukcją CIRCLE dwa poniższe wiersze:

```

LINE (314,93)-STEP(90,0)
LINE (314,93)-STEP(0,90)

```

i uruchomcie ten trzywierszowy "program". Rozkazy z LINE narysują kąt prosty w koło. Co my tu mamy? Nasze koło ma 90 pikseli szerokości, około 40 wysokości i właściwie wygląda na elipsę, a nie na koło. Coś jest nie tak. Poszukajmy zatem przyczyny.

Standardowy ekran Amigi ma, w przyjętej przez nas wielkości, wymiary 618 x 187 pikseli. A teraz zmierzmy linijką wymiary ekranu. Standardowy monitor Commodore ma 21,5 cm x 12,5 cm. Wynika z tego, że na jeden centymetr ekranu w poziomie wypada około 30 punktów, zaś na jeden centymetr w pionie około 15. A zatem stosunek wysokości do szerokości również jest (mniej więcej) jak 1:2.

Twórcy Amiga BASIC-u wiedzieli o tej sprawie i uwzględnili to przy rysowaniu. Oczywiście interpreter języka nie ma błędnego pojęcia o tym, jaki typ monitora jest podłączony do Twojej Amigi

(nie musi być to przecież koniecznie monitor "firmowy"). Czasami opisany wyżej stosunek wysokości do szerokości będzie nieco inny, i zastosowanie w instrukcji (ostatniego) parametru określającego go nie pomoże w pełni. Jeśli nie mamy innego wyjścia, należy wtedy dostosować rozmiary obrazu za pomocą odpowiednich pokręteł w monitorze, tak aby koło było rzeczywiście kołem. Ustawianie rozmiarów monitora potencjometrami nie jest jednak rozwiązaniem najlepszym, bowiem może się tak zdarzyć, że jakiś program rysujący, z którego chcesz korzystać, stosuje inne proporcje ekranowe niż BASIC. Ponadto Twój monitor może nie mieć pokręteł do regulacji wymiarów obrazu. Zastosujmy zatem inny środek polegający na ustaleniu proporcji ekranu. U nas akurat było to 1:2, czyli 0,5, u Ciebie może być nieco inaczej (przypuścimy, że po zmierzeniu i podzieleniu wyszło Ci przykładowo 0,615). Zapamiętaj tę liczbę i wszędzie tam, gdzie w instrukcji CIRCLE w naszych przykładach występuje parametr dotyczący proporcji rysunku, musisz go dostosować do swojego monitora (przy podanej wyżej wartości, tam gdzie w programie jest 0,5 dasz "swoje" 0,615, tam gdzie u nas jest 1 — Ty dasz 1,3 itd.). Wprowadźmy zatem:

```
CIRCLE (314,93),90,...,0.5
```

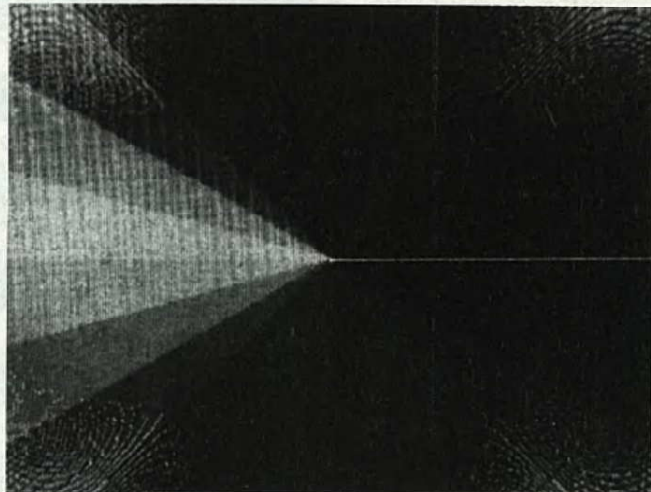
(cztery przecinki są dlatego, aby oszczędzić sobie wpisywania niepotrzebnych nam w tej chwili parametrów. Niech instrukcja przyjmie tu na razie wartości domyślne).

Uruchamiamy "program" i... Voila, oto nareszcie dokładne kółko. Oczywiście, jeśli chcieć, można zmienić dowolnie (w rozsądnych granicach) ten ostatni parametr, po to, aby uzyskać różne elipsy.

Skoro coś już umiemy, sprawdzmy to.

ĆWICZENIE 1

Sprawdźcie, metodą prób i błędów, przy jakiej wartości parametru określającego proporcje ekranu uda się za pomocą rozkazu CIRCLE nary-



Kolorowe okręgi: Moira w kolorach. Linie rysowane instrukcją CIRCLE.

```

DEF FN rad(grad)=6.283184/360*
grad

SCREEN 2,320,256,5,1
WINDOW 2,,0,2

SETRGB 1,0,0,0
SETRGB 0,15,15,15

FOR winkel=0 TO 360 STEP .2
  farbe%=INT(winkel/15)+4
  SETHSV farbe%,winkel,11,11
  COLOR farbe%,0
  CIRCLE (156,126),200,-FN rad
(winkel),-FN rad(winkel),1
NEXT winkel

WHILE INKEY$="" : WEND

WINDOW CLOSE 2
SCREEN CLOSE 2

SUB SETHSV(nr%,h,s,v) STATIC
  HSVtoRGB h,s,v,r,g,b
  PALETTE nr%,r,g,b
END SUB

SUB HSVtoRGB (h,s,v,r,g,b)
  STATIC
  r=v : g=v : b=v
  IF s>0 THEN
    hi=INT(h/60)
    hf=h/60-hi
    t1=v*(1-s)
    t2=v*(1-s*hf)
    t3=v*(1-s*(1-hf))
    IF hi=0 THEN
      g=t3 : b=t1
    ELSEIF hi=1 THEN
      r=t2 : b=t1
    ELSEIF hi=2 THEN
      r=t1 : b=t3
    ELSEIF hi=3 THEN
      r=t1 : g=t2
    ELSEIF hi=4 THEN
      r=t3 : g=t1
    ELSEIF hi=5 THEN
      g=t1 : b=t2
    END IF
  END IF
END SUB

SUB SETRGB(nr%,r%,g%,b%) STATIC
  PALETTE nr%,r%/16,g%/16,b%/16
END SUB
    
```

COLORCIRCLE Procedura SETHSV ułatwia wybór kolorów.

sować kreskę pionową, a przy jakiej pozioma.

ĆWICZENIE 2

Zaprogramujcie rysunek "Elipsy 1". Oznacza to, że macie narysować 16 elips o tym samym środku, różniących się jedynie parametrem określającym proporcje ekranu.

Wróćmy do naszego koła. W podanym wyżej przykładzie na rysowanie "okrągłego" koła

opuszczone są trzy parametry. Zgodnie ze składnią — zaraz po promieniu powinien być podany numer koloru, jakim ma być rysowane koło (domyślnie 1), następnie parametr określający "ką początkowy" (domyślnie 0) i kąt końcowy (domyślnie "dwa pi"). Oczywiście, w tym przypadku zostały przyjęte wartości domyślne.

Instrukcja CIRCLE rysuje nie tylko "całe" koła i elipsy. Za jej pomocą można także tworzyć fragmenty łuków lub wycinki koła. Jeśli chcemy to uzyskać, musimy określić parametry "kąta początkowego" i "końcowego". Wyobraźmy sobie niewidoczną linię zaczynającą się w środku koła i biegnącą w kierunku koła (na przykład lekko skośnie w lewo). Kąt, jaki tworzy ta prosta z osią poziomą X (przy założeniu, że środek układu znajduje się w środku koła,) będziemy nazywać "kątem początkowym". Druga, podobna prosta, może biec na przykład skośnie w prawo. Przez analogię możemy zatem zdefiniować "kąta końcowy". Dlaczego jednak kąty te nazywamy "początkowym" i "końcowym"? Otóż, opisane proste "wycinają" nam z koła pewien fragment, który "zaczyna" się w miejscu przecięcia z okręgiem pierwszej prostej, a "kończy" w miejscu przecięcia się z nim drugiej. Za pomocą tak zdefiniowanych wycinków możemy narysować wiele ciekawych rzeczy. Przypuśćmy, że narysujemy sobie kolistą tarczę zegara. Wystarczy teraz umówić się, że (przykładowo) kąt 0 stopni to godzina 3., 180 stopni to godzina 9., 270 stopni — 12., zaś 360 stopni — ponownie godzina 3., i już możemy rysować "wskazówki" naszego zegara. Jak wynika z powyższych wywodów — będziemy musieli w rozkazie CIRCLE podać wartości kątów. Niestety, parametrów tych nie możemy podać w stopniach. Podobnie jak w innych komputerach — wszystkie kąty należy podać w tzw. mierze łukowej, w radianach. Skąd bierze się taki radian? Przyjmuje się, że w tej mierze koło ma obwód o wielkości 6,28 jednostek. (Dokładnie jest to 2 x "liczba pi" czyli 2 x 3,141593). W mierze kątowej odpowiada to 360 stopniom. A zatem, aby

przeliczyć wartość z kąta na radiany, należy pomnożyć wielkość kątową przez 360 i podzielić przez "2 razy pi". Dla tych, którzy nie lubią bawić się w matematykę, w programie na "zegar" zastosowaliśmy funkcję przeliczającą:

```
DEFNFRAD(grad)=6.283184/360*grad
```

a następnie instrukcją:

```
rad=FNRAD(x)
```

przeliczamy wartość x podaną w stopniach na te nieszczęsne radiany.

(Od redakcji: Liczba 6,283184 to "2 * pi". Amiga BASIC nie ma liczby "pi" jako gotowej stałej. Jeśli wydaje Ci się, że powyższe przybliżenie jest zbyt mało dokładne, możesz sobie obliczyć dokładniejsze "pi" przez na przykład:

```
PI# = 4*ATN(1)
```

i do definicji funkcji zamiast 6,283184 wstawić 2*PI#)

Można także opisaną powyżej funkcję wywołującą wstawić bezpośrednio do instrukcji CIRCLE, na przykład:

```
CIRCLE (100,100),50,2,FNRAD(0),FNRAD(90)
```

jednak bardziej przejrzyste jest zastosowane przez nas rozwiązanie:

```
AW = FNRAD(0)
EW = FNRAD(45)
CIRCLE (100,100),50,2,AW,EW
```

Należy wspomnieć o jeszcze jednej zalecie rozkazu CIRCLE. Jeśli podamy ujemną wartość kąta, wówczas, oprócz utworzenia odcinka, BASIC narysuje nam linię prostą łączącą środek okręgu z początkiem (lub końcem) odcinka — tworząc w ten sposób wycinek koła. Na przykład:

```
CIRCLE (100,100),50,2,-AW,-EW
```

Po chwili eksperymentów wyniknie jednak problem. "Co będzie, jeśli któryś z kątów równy jest 0?" Przecież liczby -0 nie ma. W takim przypadku należy podać nieco mniejszą wartość, na przykład -0,00000001. Gwarantuję, że nie zauważycie różnicy w rysunku.

Instrukcja CIRCLE jest o wiele bardziej uniwersalna niż omawiana wcześniej LINE. Za pomocą CIRCLE można bowiem narysować nie tylko koło czy elipsę, ale także proste i linie. Aby zaszedł ten ostatni przypadek, oba parametry określające kąt muszą być równe. Jeśli mają wartość ujemną, wówczas zostanie narysowana linia, jeśli zaś są dodatnie — punkt.

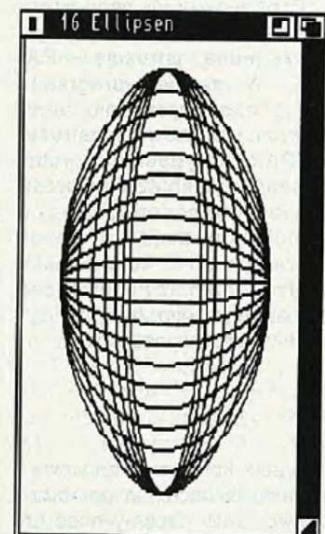
ĆWICZENIE 3

Zaprogramujcie koło rowerowe widoczne na rysunku "Elipsy 2". Nasze koło zawiera 30 szprych. Spróbujcie pierwszą z nich narysować pod kątem 6 stopni.

Teraz wiemy już wszystko, aby móc narysować nasz zegar. Spójrzcie na listing AClock. Instrukcja:

```
ON TIMER(60) GOSUB SETZEZEIT
```

zajmie się tym, aby co minutę BASIC wywołał podprogram, który najpierw wymaże wszystko, co znajduje się wewnątrz okręgu, następnie określi czas za pomocą funkcji TIME\$, przeliczy go na sekundy i minuty, a następnie (zgodnie z naszą umową przyjętą na początku) — przeliczy tak otrzymany czas na wartości kątów, które określą pozycje, w jakich będą narysowane nasze wskazówki.



Elipsy 1 Eksperymenty z proporcją ekranu.

```
DEF FN rad(grad)=6.283184/360*grad
```

```
FOR I=6 TO 360 STEP 6
  w=FN rad(I)
  CIRCLE (300,100),150,,w,.5
NEXT I
FOR I=30 TO 360 STEP 30
  w=FN rad(I)
  CIRCLE (300,100),150,,w,.5
NEXT I
GOSUB SetzeZeit
```

```
ON TIMER(60) GOSUB SetzeZeit
TIMER ON
WHILE INKEY$=""
  SLEEP
WEND
CLS
END
```

```
SetzeZeit:
  CIRCLE (300,100),140,2,,.5
  PAINT (200,100),0,2
  s=VAL(LEFT$(TIME$,2))
  m=VAL(MID$(TIME$,4,2))
  s=(s+9) MOD 12 : s=s+m/60
  m=(m+45) MOD 60
  s=12-s : m=60-m
  sw=6.283184/12*s : mw=6.283184/60*m
  CIRCLE (300,100),140,,mw,mw,.5
  CIRCLE (300,100),120,,sw,sw,.5
RETURN
```

ACLOCK Przykład zastosowania instrukcji CIRCLE.

Składnia rozkazu PAINT

PAINT (x,y), kolor wypełnienia, kolor obrzeża.

Przy okazji wpisywania programu AClock poznamy jeszcze jedną instrukcję — PAINT. W naszym programie maże ona poprzednio narysowane wskazówki. Instrukcja PAINT wypełnia dowolny obszar zamknięty kolorem, rozpoczynając od punktu o współrzędnych (x,y). Można także określić kolor, jakim będzie namalowany brzeg zamalowywanego obszaru. Przykładowo zestaw instrukcji:

```
CIRCLE (300,100),50,2,,.5
PAINT (300,100),2
```

narysuje koło pomarańczowe i wypełni je także na pomarańczowo. Jeśli chcemy mieć brzeg innego koloru, musimy dodatkowo określić jego kolor. Przykładowo ostatnią instrukcję można poprawić na:

```
PAINT (300,100),23
```

i w ten sposób otrzymamy pomarańczowe koło z czarną obwódką.

Podczas rysowania koła lub elipsy kursor graficzny znajduje się w jego środku. Można zatem wykorzystać (jeżeli nie zapomnieliście tego) omawiane w poprzednim odcinku kursu współrzędne "relatywne" i parametr STEP, na przykład:

```
PAINT STEP (0,0),2
```

Przy posługiwaniu się rozkazem PAINT należy mieć na uwadze jedno. Sposób pracy tej instrukcji można przyrównać do nalewania wody do naczynia. Jeśli ma ono dziurę — wszystko się wyleje. Podobnie, jeśli PAINT znajdzie "dziurę" w zamalowywanym obszarze (tzn. nie będzie on całkowicie zamknięty) — też się "wyleje" i w efekcie najczęściej zostanie zamalowany cały ekran. Wystarczy jeden piksel nieszczelności. Ponadto — punkt, od którego rozpoczynamy zamalowywanie, MUSI znajdować się wewnątrz zamalowywanego obszaru. W przeciwnym razie zamalowane zostanie nie to, co chcieliśmy zamalować.

Wskazówki naszego zegara nie będą niestety identyczne. Dlaczego? Bierze się to z nieszczęsnych proporcji ekranu. Skoro centymetr mierzony "w pionie" zawiera o połowę mniej punktów niż ten

"dotychczasowa" rozdzielczość 640x256. Użyte powyżej pojęcie "rozdzielczości" nie jest prawdziwe, bowiem zagęszczenie punktów, mimo iż wyrażone cyfrowo (na przykład 640 x 400) będzie takie samo — inaczej wygląda na dużym monitorze, a inaczej na małym. Pozostawimy jednak przy tym popularnym określeniu.

Składnia instrukcji SCREEN

SCREEN numer, szerokość, wysokość, "głębokość", tryb graficzny.

Powierzchnia, na której wyświetlane są obrazy Amigi nazywa się "ekranem" (ang. screen). Nie należy jednak mylić tego pojęcia z ekranem monitora. Rozkaz SCREEN obsługuje poszczególne ekrany. Parametrami są tu: kolejny numer ekranu, jego wysokość i szerokość, parametr określający liczbę kolorów (nazwany przez twórców, "głębokością"), który będzie omówiony za chwilę. Ostatni parametr określa tryb graficzny, w jakim będzie pracował dany ekran. W poniższej tabelce opisane są tryby graficzne Amigi dostępne z poziomu Amiga BASIC.

Nazwy trybów graficznych w tabelce biorą się z połączenia angielskich słów low (niski), high (wysoki) i res[olution] (rozdzielczość). PAL (Phase Alternation Line) i NTSC (National Television Standard Code) — to systemy telewizyj-

Przecież trzeci parametr określa kolory. Mamy mieć cztery, a nie dwa. Racja, tyle tylko że parametr nazywa się prześmiesznie "głębokość", a nie liczba kolorów. Podstawiana tu wartość wynika z niuansów kodu dwójkowego. Zasadą jest, że dwa podniesione do potęgi, podanej jako ten parametr, daje nam liczbę kolorów. Ponieważ bez sztuczek (na które przyjdzie czas na końcu kursu) — w Amiga BASIC-u można używać minimalnie dwóch, a maksymalnie trzydziestu dwóch kolorów, jasnym jest, że parametr "głębokość" może przyjmować wartości od 1 (2¹) do 5 (2⁵). Przy ustalaniu wartości tego parametru musicie jednak pamiętać o tym, że w trybie ze zwiększoną liczbą punktów pionowych można w BASIC-u korzystać maksymalnie z sześciastu kolorów.

Składnia instrukcji WINDOW

WINDOW numer, nazwa, (x1,y1)-(xp,yp), flaga, numer ekranu.

Aby w programie BASIC można było przedstawić wyniki działania instrukcji graficznych, musimy otworzyć okienko. Robi to za nas instrukcja WINDOW. Ponieważ w programie możemy otworzyć kilka okien jednocześnie, zatem pierwszym parametrem jest numer okna. (Do numeru tego będziemy się odwoływać gdy zajdzie taka potrzeba.) Aby umieścić okno na ekranie, musimy, podobnie jak przy rysowaniu prostokąta za pomocą instrukcji LINE, określić współrzędne lewego górnego i prawego dolnego narożnika okienka. UWAGA: Jeśli nasze okno będzie miało listwę tytułową, wówczas górną współrzędną Y musimy zwiększyć o co najmniej 8. Dlaczego zwiększyć? Przypomnij sobie, w którą stronę skierowana jest oś pionowa Y. Dobrze jest też (w przypadku otwierania okna na cały ekran) nie "dociągać" z oknem do samego końca, a zatem zamiast (640,200) bezpieczniej będzie podać (639,199). Różnicy nie zauważycie, a unikniecie kilku kłopotów. Pa-

TRYBY GRAFICZNE AMIGA BASIC

Tryb graficzny	Rozdzielczość	Nazwa trybu
1	320 x 256	PAL - Lores
2	640 x 256	PAL - Hires
3	320 x 512	Pal - Lores - Interlance
4	640 x 512	PAL - Hires

sam centymetr mierzony "w poziomie", to muszą występować różnice w wyglądzie, bowiem nasze wskazówki (zgodnie z definicją okręgu) muszą mieć zawsze tę samą długość, bez względu na położenie.

Jak ustawić wielkość ekranu na Amidze? Okno, z jakiego dotąd korzystaliśmy, jest związane z Workbenchem. Stąd

zyczne używane w Niemczech i USA. Przypuśćmy, że chcemy ustawić ekran, który będzie miał rozdzielczość 320 x 256 punktów i 4 kolory w trybie PAL Lores. Odpowiednia instrukcja wygląda tak:

```
SCREEN 2,320,256,2,1
```

"Zaraz, zaraz!" — zaprotestuje nasz uważny Czytelnik.



parametr "flaga" pozwala na dotarcie do okna różnych symboli ułatwiających jego obsługę (tzw. gadżetów). Wartości tego parametru przedstawiono w tabelce poniżej.

GADŻETY OKIEN AMIGI	
parametr "flaga"	na ekranie pojawi się gadżet umożliwiający:
0	(okno bez gadżetów)
1	zmianę wymiarów okna myszką
2	przesuwanie okna po ekranie
4	chowanie lub wysuwanie okna
8	usunięcie okna z ekranu
16	przesuwanie zawartości okna

Uwaga: Jeśli chcemy uzyskać kilka gadżetów – należy zsumować odpowiadające im wartości

Do naszych eksperymentów na razie nie potrzebujemy gadżetów, a zatem podamy tu wartość 0.

Numer ekranu określa to, na jakim ekranie pojawi się nasze okno graficzne. Jeśli go opuścimy lub podamy domyślną wartość 1, wówczas nasze okno pokaże się na ekranie Workbench. Z tego powodu należy tu podawać numer różny od 1.

Teraz nauczymy się pewnej sztuczki. Jeśli podacie 0 jako parametr "flagi" i jednocześnie "opuścicie" parametr "nazwa", wówczas nasze okno graficzne pojawi się bez listwy tytułowej i gadżetów, co nieraz jest efektem pożądanym. Jeśli natomiast opuścicie współrzędne określające położenie narożników okna, wówczas okno graficzne będzie miało maksymalne, możliwe w danej rozdzielczości, rozmiary (to znaczy otworzy się "na całym ekranie"). Każdy otwarty ekran i okno należy zamknąć wówczas, gdy nie zamierzamy się więcej do niego odwoływać. Wprowadźcie w naszym przykładowym programie nie będzie to konieczne, jednak zalecam Wam, abyście to zrobili. Aby zamknąć ekran (lub okno), wystarczy użyć tej samej instrukcji, jaką ekran (okno) został otwarty, zaś jedynymi parametrami tej instrukcji będą słowo CLOSE i numer zamykanego ekranu (okna), na przykład SCREEN CLOSE 2.

Obszar rysowania określmy sobie teraz za pomocą dwóch

podprogramów. W programie VIEWS podprogram OPENVIEW służy do określania numeru ekranu i okna, tytułu okna, liczba stosowanych kolorów (nie ma tu żadnej stałej liczby, lecz

parametr jest wartością zmienną) i tryb graficzny, w jakim chcemy pracować. CLOSEVIEW zamyka ekran i okno.

A teraz znów nieco ćwiczeń:

ĆWICZENIE 4

Zaprogramujcie Basicowy zegar w rozdzielczościach 320 x 256 i 640 x 512.

ĆWICZENIE 5

Okreście dla każdej rozdzielczości proporcje ekranu (przypominam, że jest to ostatni parametr rozkazu CIRCLE). Podstawcie tę wartość pod zmienną bv. Najprostszą chyba metodą rozwiązania problemu będzie uzupełnienie instrukcji IF z podprogramu OPENVIEW o parametr "wysokość". Gdy potem jeszcze dopiszecie wiersz

SHARED bv

poniżej instrukcji SUB OPENVIEW, wówczas parametr bv będzie można stosować do dowolnego koła i wynikiem działania rozkazu CIRCLE będzie zawsze idealnie okrągłe kółko, bez względu na zastosowaną rozdzielczość.

Znamy już zasady działania rozkazów rysujących Amiga BASIC-u i poznaliśmy praktyczne efekty ich działania. Postaramy się umieścić je wszystkie w nieco bardziej rozbudowanym programie rysunkowym BASIC DRAW. Za pomocą tego programu można będzie interaktywnie rysować takie elementy grafi-

czne jak linia, kwadrat, łuk lub zamalowywać zamknięte obszary. W rysowaniu okręgów za pomocą myszy przydadzą się funkcje trygonometryczne: sinus, cosinus i tangens. Rozkazy "wypełniające" również obsługujemy tym sympatycznym zwierzątkiem. Wystarczy "kliknąć", gdy wskaźnik znajduje się w miejscu, gdzie chcemy umieścić współrzędne początkowe (lub środek okręgu), następnie przesunąć myszkę do miejsca współrzędnych końcowych (obwodu koła) i ponownie "tupnąć". Podczas poruszania myszki będziecie widzieć zmieniający się kształt elementu. Klawiszem [L] uruchomicie "narzędzie" pozwalające rysować linie, odpowiednio [R] uaktywni rysowanie prostokątów, zaś [K] koła. Klawisz funkcyjny [F1] zmieni kolor, jakim rysowane będą elementy, a klawisz [F2] — kolor tła. To ostatnie zmieni również kolor (na domyślny), gdy będziecie mazać swoje arcydzieło, uży-

```
SUB OPENVIEW(nr%,title$,modus%
,anzf%) STATIC
IF modus%=1 THEN b=320 : h=
256
IF modus%=2 THEN b=640 : h=
256
IF modus%=3 THEN b=320 : h=
512
IF modus%=4 THEN b=640 : h=
512
t=CINT((LOG(anzf%)/LOG(2)))
SCREEN nr%,b,h,t,modus%
WINDOW nr%,title$,0,nr%
END SUB
```

```
SUB CLOSEVIEW(nr%) STATIC
SCREEN CLOSE nr%
WINDOW CLOSE nr%
END SUB
```

VIEWS Łatwe zarządzanie screenami.

wając w tym celu klawisza [Del]. Sądzę, że powinniście mieć sporo satysfakcji z tego pierwszego "poważniejszego" programu. Oczywiście nic nie stoi na przeszkodzie, abyście po sporządzeniu na wszelki wypadek kopii dokonywali najróżniejszych zmian w tym kolorze. Eksperymentując możecie oprzeć się na tym, czego dotąd się nauczyliście, jak i na wiedzy, którą zdobędziecie w dalszym ciągu kursu.

Składnia instrukcji COLOR

COLOR — kolor rysowania, kolor podkładu.

Kolor można wprowadzać w poznanych już instrukcjach graficznych. Można w tym celu użyć także instrukcji COLOR. Jeśli używamy znaków, wówczas kolor rysowania (także przy PRESET) nie ma znaczenia. Jeśli jednak, przykładowo, użyjemy rozkazu CLS (Clear screen — czyść ekran), wówczas ekran zostanie zamalowany aktualnym kolorem podkładu. Proponowany przez nas program BASIC DRAW zawiera podprogram o nazwie DRAWMODE, który ustala tryb rysowania. Szerzej zostanie on omówiony w kolejnej części kursu. Podprogram ułatwiający interaktywne rysowanie nazywa się DRAWMODE2. Włącza on tryb XOR. Oznacza to, że rysowane elementy są słabiej widoczne i dopiero w momencie "zatwierdzenia" kształtu przez powtórne naciśnięcie myszki tryb XOR zostanie wyłączony i narysowany element będzie wyraźnie widoczny. Jednak procedura ta używa najbardziej "niebezpiecznej" instrukcji Amiga BASIC-u, a mianowicie POKE. Podczas wpisywania tego podprogramu należy bardzo uważać. Jedną pomyłką w parametrze POKE może (choć nie musi) spowodować to, że nasza Amiga "zwariuje". W dalszej części kursu pokażemy, w jaki sposób zabezpieczyć się przed konsekwencjami błędnego POKE.

Poeksperymentujmy sobie trochę. Dopiszcie podprogram DRAWMODE2 do poznanego w pierwszej części kursu programu na "linie losowe" i zobaczcie, co z tego wyniknie (pod warunkiem, że na końcu tego programu nie zapomnicie dopisać podprogramu DRAWMODE1). Program BASIC DRAW będzie wymagał od Was nieco więcej pracy przy wpisywaniu. Jednak nie jest naszym głównym celem to, abyście go jedynie wpalowali i używali. Program prezentuje nam różne metody grafiki interaktywnej. Te umiejętności mogą się później bardzo przydać (nawet, jeśli porzucicie

```

SCREEN 2,320,256,5,1
WINDOW 2,,,,2

bv.=1
vgf.%=1 : hgf.%=0 : rf.%=1

Werkzeug=1
ON MOUSE GOSUB BearbeiteMaus :
MOUSE ON

WHILE Ende=0
dummy=MOUSE(0)
mx=MOUSE(1) : my=MOUSE(2)
e$=INKEY$
IF e$ <> "" THEN GOSUB SetzeW
erkzeug
ON Werkzeug GOSUB Linie,Linie
,Kreis,Kreisbogen,
Fuellen,Text
WEND

WINDOW CLOSE 2
SCREEN CLOSE 2
END

BearbeiteMaus:
klick=klick+1
IF klick=1 THEN
mx1=mx : my1=my
END IF
RETURN

SetzeWerkzeug:
e=ASC(e$)
IF e$="1" THEN Werkzeug=1
IF e$="r" THEN Werkzeug=2
IF e$="k" THEN Werkzeug=3
IF e$="b" THEN Werkzeug=4
IF e$="f" THEN Werkzeug=5
IF e$="t" THEN Werkzeug=6
IF e=27 THEN klick=0
IF e=127 THEN CLS
IF e=129 THEN vgf.%=VAL(zahl
$) : COLOR vgf.%,hgf.%
IF e=130 THEN hgf.%=VAL(zahl
$) : COLOR vgf.%,hgf.%
IF e=131 THEN rf.=VAL(zahl$)
IF e=138 THEN Ende=1
IF e=134 THEN h.=VAL(zahl$)
: SETHSV vgf%,h.,s.,v.
IF e=135 THEN s.=VAL(zahl$)
: SETHSV vgf%,h.,s.,v.
IF e=136 THEN v.=VAL(zahl$)
: SETHSV vgf%,h.,s.,v.
IF (e$>="0" AND e$<="9") OR
e$="." THEN
zahl$=zahl$+e$
ELSE
zahl$=""
END IF
RETURN

Linie:
IF klick <> 0 THEN
DRAWMODE 2
IF Werkzeug=1 THEN
LINE (mx1,my1)-(mx,my)
ELSE
LINE (mx1,my1)-(mx,my),,b
END IF
IF klick > 1 THEN
klick=0
DRAWMODE 1
END IF
IF Werkzeug=1 THEN
LINE (mx1,my1)-(mx,my)
ELSE
LINE (mx1,my1)-(mx,my),.
END IF
END IF

```

```

RETURN

Kreis:
IF klick <> 0 THEN
DRAWMODE 2
radius=ABS(mx-mx1)
CIRCLE (mx1,my1),radius,,,
,bv.
IF klick > 1 THEN
klick=0
DRAWMODE 1
END IF
CIRCLE (mx1,my1),
radius,,,bv.
END IF
RETURN

Fuellen:
IF klick <> 0 THEN
PAINT (mx1,my1),,rf.
klick=0
END IF
RETURN

Text:
IF klick <> 0 THEN
LOCATE INT(my1/6)+1,INT(m
x1/8)+1
INPUT "",dummy$
klick=0
END IF
RETURN

SUB DRAWMODE(mode%) STATIC
POKE WINDOW(8)+28,mode%
END SUB

SUB SETHSV(nr%,h,s,v) STATIC
HSVtoRGB h,s,v,r,g,b
PALETTE nr%,r,g,b
END SUB

SUB SETRGB(nr%,r%,g%,b%) STATIC
PALETTE nr%,r%/16,g%/16,b%/16
END SUB

SUB RGBtoHSV(r,g,b,h,s,v)
STATIC
v=r
IF v < g THEN v=g
IF v < b THEN v=b
min=r
IF min > g THEN min=g
IF min > b THEN min=b
ba=v-min
h=-1 : s=0
IF v > 0 AND v < 1 THEN
s=ba/v
IF v=r THEN
h=(g-b)/ba
ELSEIF v=g THEN
h=(b-r)/ba+2
ELSEIF v=b THEN
h=(r-g)/ba+4
END IF
IF h < 0 THEN h=h+6
h=h/60
END IF
END SUB

SUB HSVtoRGB (h,s,v,r,g,b)
STATIC
r=v : g=v : b=v
IF s > 0 THEN
hi=INT(h/60)
hf=h/60-hi
t1=v*(1-s)
t2=v*(1-s+hf)
t3=v*(1-s*(1-hf))
IF hi=0 THEN
g=t3 : b=t1
ELSEIF hi=1 THEN

```

```

r=t2 : b=t1
ELSEIF hi=2 THEN
r=t1 : b=t3
ELSEIF hi=3 THEN
r=t1 : g=t2
ELSEIF hi=4 THEN
r=t3 : g=t1
ELSEIF hi=5 THEN
g=t1 : b=t2
END IF
END IF
END SUB
BASIC DRAW Inter-
aktywne rysowanie elemen-
tów graficznych.

```

BASIC i "przeziądziecie" się na programowanie, powiedzmy w języku C). Przeanalizujcie zatem gruntownie jego listing.

* Czy pamiętacie nasz "ploter" z poprzedniego odcinka kursu? Za pomocą podprogramów MOVETO, DRAWTO i PLOTBJECT rysowaliśmy sobie litery na ekranie. Dokonując w tym programie niewielkich zmian, możemy uzupełnić go o nowe możliwości.

Składnia instrukcji AREA

AREA (x,y)
Przypuśćmy, że przyszła nam na przykład ochota na to, by pomalować litery narysowane za pomocą PLOTBJECT. Rozkaz PAINT będzie tu mało przydatny, bowiem do rysowania używaliśmy współrzędnych związanych z daną literą, a przy tak nieregularnych kształ-

tach "wyciągnięcie" współrzędnych z powrotem będzie benedyktyńskim mozolem. Pomyślcie ponadto, ile to będzie przepisywania i dopisywania. Użyjemy instrukcji AREAFILL. Instrukcja ta wypełnia obszar określony wcześniej instrukcją AREA. W naszym programie PLOTBJECT pobiera współrzędne z wierszy danych i przekazuje je do DRAWTO i MOVETO. Rozszerzmy oba te podprogramy o instrukcję:

```

AREA (x,window(3)-y)

A jak będzie wypełniany obszar? W tym celu przed PLOTBJECT należy dopisać poniższe wiersze:
....
NEXT i
AREAFILL
READ n
....

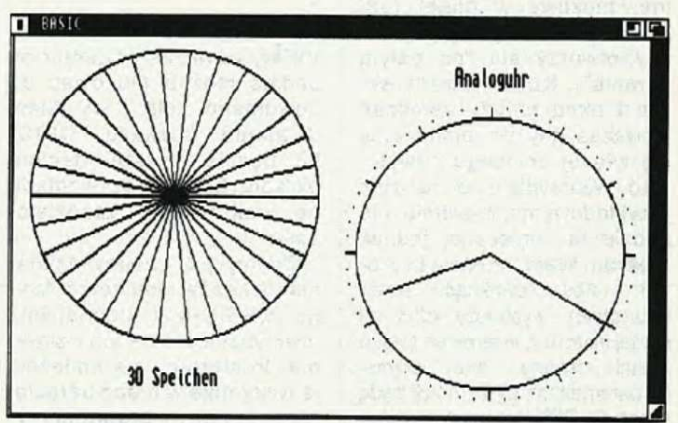
```

To już mamy. Ale jakim kolorem będą wypełniane nasze litery? AREAFILL przyjmuje ustawioną instrukcją COLOR barwę tła. Nasza litera A składa się z dwóch powierzchni. Obie zostałyby wypełnione tym samym kolorem. Tak nie może być. Kolor dla każdego obszaru musimy umieścić w wierszach danych. Przykładowo możemy poprawić dane następująco (porównaj poniższe wiersze z odpowiednimi wierszami "starego" programu):

```

LiteraA:
DATA 17,2,3,1,3,3,...
DATA 9,6,0,12,7,0,...
Numer będzie czytany razem

```



Elipsy 2 Instrukcja CIRCLE rysuje elipsy, koła i okręgi (program A Clock steruje zegarem).



ze współrzędnymi. Aby jednak powiększenie liczby danych nie spowodowało zamieszania, musimy także odpowiednio poprawić instrukcję READ, jak poniżej:

```
READ x,y,I
```

Teraz wystarczy tylko przed instrukcją AREAFILL umieścić instrukcję:

```
COLOR I,0
Gotowe.
```

* Przypuśćmy, że udało Wam się stworzyć jakąś przyzwoitą grafikę. Ale co za pożytek z tego, jeśli po wyłączeniu komputera można o niej zapomnieć. Programiści z Microsoftu zapomnieli w Amiga BASIC-u umieścić rozkazy pozwalające na zapisywanie i odczytywanie obrazków. Nie jest jednak tak źle. Opiszemy to w następnym odcinku kursu (a może nawet poświęcimy temu tematowi jakiś oddzielny artykuł — przyp.

red.). Przypominam o potrzebie eksperymentowania i życzę wielu sukcesów w programowaniu. Macie na to cały miesiąc.

AMIGA Magazin 9/91
Ilum. Marek Pampuch

Literatura w języku niemieckim:

1. Harald Kupers: Harmonielehre der Farben; Du Mont, 1989; ISBN 3-7701-2192-9
2. Das neue Supergrafikbuch; Data Becker Verlag, 1988; ISBN
3. W. D. Fellner: Computergrafik; BI-Wissenschaftsverlag, 1988; ISBN 3-411-03179-4

4. Foly Dam/Feiner/Hughes: Computer Graphics; Addison Wesley, 2. Aufl. 1990; ISBN 0-201-12110-7

5. Markus Breuer: Faszination der Computergrafik; AMIGA-Magazin 3/89, Seite 84

6. Peter Schone: Phänomen Farben, AMIGA-Magazin 11/89, Seite 112

7. Axel Bruck: Computergrafik; Falken Verlag, 1987; ISBN 3-8068-4319-8

8. Norbert Spittenardt: Frischzellen für Basic II; AMIGA-Magazin 7/91, Seite 140

Literatura w języku polskim:

9. "Moja Amiga" cz. II — Amiga BASIC — Fundacja Edukacji Technologicznej, Warszawa 1991

Marek Pampuch

Zgodnie z obietnicą — zamieszczamy rozwiązania ćwiczeń z pierwszej części kursu programowania grafiki w BASIC-u. Oczywiście nie są to rozwiązania jedyne. BASIC jest na tyle elastyczny, że ten sam efekt możesz uzyskać w inny sposób. Nie twierdzimy też, że są to rozwiązania najlepsze. Jeśli uznasz, że Twój "sposób na dane ćwiczenie" jest lepszy — nadesłaj listing na adres redakcji. Najlepsze rozwiązania zamieścimy, wynagradzając według obowiązujących stawek.

UWAGA: Wszystkie programy oraz rozwiązania zadań kursu znajdują się (oprócz innych programów) na pierwszej dyskietce dla prenumeratorów, która ukaże się w grudniu. Osoby nie prenumerujące Magazynu AMIGA będą mogły nabyć tę dyskietkę na warunkach, jakie podamy w następnym numerze.

ĆWICZENIE 1

Zadanie wydaje się banalnie proste. Okazuje się jednak, że najprostsze rzeczy bywają właśnie najtrudniejsze. W momencie przystępowania do wykonywania ćwiczenia "nie znamy" jeszcze rozkazu LINE, a zatem musimy użyć PSET. Najpierw ustalamy współrzędne punktu, z którego będą rysowane linie.

Przykładowo:

```
X=160:Y=100
```

Teraz procedura rysowania "pionowo w górę":

Rozwiązania ćwiczeń z 1 części

```
X1=X:Y1=Y
5 Y1=Y1-1:IF Y1<50 THEN 10
PSET (X1,Y1),1
GOTO 5
10 REM pionowo w dół
```

Zobaczmy, co tu się dzieje. Najpierw ustalamy współrzędne "robotce" dla instrukcji PSET. Nie może to być X i Y, bowiem współrzędne te będą się w trakcie rysowania zmieniać, a skoro wszystkie proste mają wyjść z tego samego punktu, to współrzędne "początkowe" jeszcze nam się przydadzą. W związku z tym współrzędne robotce określamy jako X1 i Y1. Na początku są one równe punktowi początkowemu. Współrzędna X1 właściwie tutaj nie będzie nam potrzebna, ale dla ułatwienia sobie dalszej pracy zostawiamy ją. W wierszu z etykietą 5 zmniejszamy Y (bo linia ma iść pionowo w górę) o 1. Sprawdzamy, czy osiągnęliśmy koniec (w rozwiązaniu przyjęłem długość wszystkich linii prostych równą 50 — po osi X i po osi Y. Wy oczywiście możecie przyjąć inne wartości, starając się nie wyjść poza obręb okienka). Jeśli do końca nam daleko, wówczas warunek po IF nie jest spełniony i program przechodzi do nowego wiersza, w którym rysuje za pomocą

PSET kolejny punkt prostej, a następnie wraca do wiersza z etykietą 5 i tak "w koło Maciejju", aż "przekroczymy" koniec (tu Y=100-50, czyli 50), wówczas nasze ciągle zmniejszane Y wyniesie 49 i warunek po IF zostanie spełniony, a zatem program przejdzie do wiersza z etykietą 10. Wiersz ten nie należy już (jak wskazuje komentarz) do rysowania prostej "w górę", ale znów dla wygody tu go umieszczam.

Na czym polega wspomniana dwukrotnie "wygoda"? Wystarczy teraz oznaczyć powyższą procedurę za pomocą myszy jako blok, skopiować go (COPY z listwy tytułowej), ustawić kursor w pierwszym wolnym wierszu i "dokleić" blok (za pomocą PASTE z listwy tytułowej). Ale to jeszcze nie wszystko. Kolejna prosta ma być rysowana "w dół" a zatem współrzędna robocza będzie się powiększać do (100+50). Trzeba także odpowiednio pozmienić nazwy etykiet i odwołania do nich. Można też (choć to niekonieczne) zmienić kolor rysowanej linii odpowiednim parametrem w PSET. Po skopiowaniu i poprawkach kolejny fragment będzie wyglądał mniej więcej tak:

```
X1=X:Y1=Y
```

```
15 Y1=Y1-1:IF Y1<150 THEN 20
PSET (X1,Y1),2
GOTO 15
20 REM poziomo w prawo
```

I tak jeszcze sześć razy. Po ostatnim bloku (jeśli zachowasz numerację etykiet jak tu, wówczas po narysowaniu ostatniej z wymaganych prostych nastąpi skok do etykiety 80) dobrze jest zastosować "pętlę czasową", abyśmy mogli podziwiać nasz rysunek jeszcze około 5 sekund po narysowaniu:

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

I to już wszystko. Cały "program" wygląda tak:

```
X=160:Y=100
REM PIONOWO W GÓRĘ
X1=X:Y1=Y
5 Y1=Y1-1:IF Y1<50 THEN 10
PSET (X1,Y1),1
GOTO 5
10 REM pionowo w dół
X1=X:Y1=Y
15 Y1=Y1-1:IF Y1>150 THEN 20
PSET (X1,Y1),2
GOTO 15
20 REM poziomo w prawo
X1=X:Y1=Y
25 X1=X1+1:IF X1>210 THEN 30
PSET (X1,Y1),3
GOTO 25
30 REM poziomo w lewo
X1=X:Y1=Y
35 X1=X1-1:IF Y1<110 THEN 40
PSET (X1,Y1),1
GOTO 35
40 REM ukośnie prawo-góra
X1=X:Y1=Y
45 Y1=Y1-1:X1=X1+1:IF X1>210 THEN 50
PSET (X1,Y1),2
GOTO 45
50 REM ukośnie lewo-góra
X1=X:Y1=Y
```

```
55 Y1=Y1-1:X1=X1-1:IF Y1<110 THEN
60
PSET (X1,Y1),2
GOTO 55
60 REM ukośnie lewo-dół
X1=X:Y1=Y
65 Y1=Y1+1:X1=X1-1:IF X1<110 THEN
70
PSET (X1,Y1),3
GOTO 65
70 REM ukośnie lewo-góra
X1=X:Y1=Y
75 Y1=Y1-1:X1=X1+1:IF X1>150 THEN
80
PSET (X1,Y1),1
GOTO 75
80 FOR I=1 TO 5000:NEXT I
```

ĆWICZENIE 2

Tu wpisywania będzie znacznie mniej, ale za to nie tak "bezmyślnego" jak w ćwiczeniu 1. Choćby na przykładzie tych programów widzicie, jak czasem pomysł oszczędzi niepotrzebnej pracy. Przypominam, że ćwiczenie polegało na narysowaniu linii "punktowanej". Oto przykładowe rozwiązanie:

Określamy współrzędne początkowe (x,y), "odstęp", w jakim mają być rysowane punkty, i długość linii (d). Można też (choć nie jest to konieczne) określić kolor rysowania (k):

```
X=100:Y=50:Z=10:D=200:K=1
```

Teraz pozostało już tylko rysowanie:

```
FOR i=x TO x+d STEP z
PSET (i,y),k
NEXT
```

i to wszystko.

Powyższy przykład narysuje linię poziomą. Spróbujmy teraz narysować linię pionową, zmieniając odstęp między punktami. Poprawek będzie niewiele:

```
X=100:Y=150:Z=5:D=100:K=2
FOR i=y TO y+d STEP z
PSET (x,i),k
NEXT
```

Spróbujcie teraz jako wprawkę narysować linię ukośną. Tu już nie dam wam "ściągali".

ĆWICZENIE 3

Ćwiczenie polegało na odtworzeniu "rastra" z rysunku w artykule. Sprawa nie jest trudna. Wystarczy "odpowiednio" połączyć oba powyższe "programy" i dopasować parametry.

W efekcie wyjdzie coś takiego:

```
x=10:y=10
FOR i=x TO 120 STEP 20
FOR j=y TO 50 STEP 10
PSET (i,j)
NEXT j
NEXT i
```

Różne "długości" po osi X i Y, oraz różne "kroki" wynikają z proporcji ekranu.

ĆWICZENIE 4

Aby program "Linie losowe" nie rysował "w kolorze zerowym", wystarczy odpowiednio ograniczyć zakres uzyskiwanych funkcją RND wartości. Można to zrobić na przykład tak:

```
f=RND*2
LINE (x1,y1)-(x2,y2),f+1
```

Pozostałe wiersze pozostają bez zmian. W pierwszym zmienianym wierszu generujemy liczby z zakresu 0-2. Ponieważ jednak nasz parametr f ma mieć wartość z zakresu 1-3, w następnej linii zwiększamy go o 1.

ĆWICZENIE 5

Tu chodziło o raster "liniowy" z rysunku w artykule. Wbrew pozorom nie można tego zrobić przez proste zastąpienie rozkazu PSET z ćwiczenia 3. przez rozkaz LINE. Jeśli nie wierzysz — spróbuj. Należy rozdzielić rysowanie linii poziomych i pionowych, na przykład tak jak poniżej:

najpierw ustalamy współrzędne "narożników" naszego pokratkowanego prostokąta:

```
x1=10:y1=10:x2=110:y2=50
```

teraz rysujemy linie poziome (to znaczy te, w których zmieniła się tylko współrzędna X):

```
FOR i=x1 TO x2 STEP 20
LINE (i,y1)-(i,y2)
NEXT i
```

a teraz to samo dla linii pionowych:

```
FOR i=y1 TO y2 STEP 10
LINE (x1,i)-(x2,i)
NEXT i
```

i to już wszystko. "Dla porządku" cały program:

```
x1=10:y1=10:x2=110:y2=50
FOR i=x1 TO x2 STEP 20
LINE (i,y1)-(i,y2)
NEXT i
FOR i=y1 TO y2 STEP 10
LINE (x1,i)-(x2,i)
NEXT i
```

ĆWICZENIE 6

Należało narysować sześciąt, z przerywanymi liniami, które są zastłonięte. Najpierw wpisujemy procedurę "setpattern", która pozwoli nam uzyskać linie przerywane (przepisujemy sam podprogram, bez trzech ostatnich wierszy demonstrujących działanie programu). Ale to jednak nie wszystko. Teraz weźmy kawałek kartki i narysujmy na niej nasz sześciąt. Ponumerujmy wierzchołki i łączące je linie i opiszmy współrzędne wierzchołków. Ja przyjąłem następujące:

```
x1=10:y1=20
x2=160:y2=20
x3=160:y3=90
x4=10:y4=90
x5=30:y5=10
x6=180:y6=10
x7=180:y7=80
x8=30:y8=80
```

Wpiszmy te współrzędne do komputera. Zostało nam jeszcze tylko narysowanie sześciąt. Najpierw narysujmy linie "widoczne":

```
setpattern ***
LINE (X1,Y1)-(X2,Y2)
LINE (X2,Y2)-(X3,Y3)
LINE (X3,Y3)-(X4,Y4)
LINE (X1,Y1)-(X4,Y4)
LINE (X1,Y1)-(X5,Y5)
LINE (X5,Y5)-(X6,Y6)
LINE (X6,Y6)-(X2,Y2)
LINE (X6,Y6)-(X7,Y7)
LINE (X3,Y3)-(X7,Y7)
```

Uważny Czytelnik zwróci uwagę na to, że niektóre linie są rysowane "w drugą stronę". Po prostu tak było wygodniej wpisywać.

Teraz narysujemy przerywane linie "niewidoczne":

```
setpattern ***
LINE (X4,Y4)-(X8,Y8)
LINE (X8,Y8)-(X7,Y7)
LINE (X8,Y8)-(X5,Y5)
```

Teraz jeszcze tylko ustawmy wzorec rysowania linii z po-

wrotem na linię ciągłą, aby przy kolejnym ćwiczeniu nie spotkała nas niespodzianka:

setpattern ***
i możemy już spojrzeć na cały program:

```
REM tu ma być podprogram SETPATTERN
```

```
x1=10:y1=20
x2=160:y2=20
x3=160:y3=90
x4=10:y4=90
x5=30:y5=10
x6=180:y6=10
x7=180:y7=80
x8=30:y8=80
setpattern ***
LINE (X1,Y1)-(X2,Y2)
LINE (X2,Y2)-(X3,Y3)
LINE (X3,Y3)-(X4,Y4)
LINE (X1,Y1)-(X4,Y4)
LINE (X1,Y1)-(X5,Y5)
LINE (X5,Y5)-(X6,Y6)
LINE (X6,Y6)-(X2,Y2)
LINE (X6,Y6)-(X7,Y7)
LINE (X3,Y3)-(X7,Y7)
setpattern ***
LINE (X4,Y4)-(X8,Y8)
LINE (X8,Y8)-(X7,Y7)
LINE (X8,Y8)-(X5,Y5)
setpattern ***
```

ĆWICZENIE 7

W tym ćwiczeniu chodziło o to, aby na ekranie wypisywane były współrzędne punktu, w którym znajduje się aktualnie wskaźnik. Aby współrzędne były wypisane — należy nacisnąć lewy klawisz myszki. Jeden rzut oka na program MiniPaint i już widzimy, że wystarczy go tylko nieco zmodyfikować, na przykład tak jak poniżej:

```
WINDOW 1, ""
loop:
WHILE MOUSE(0)=0 AND WINDOW(0)=1
WEND
IF WINDOW(0)=1 THEN CLS:PRINT
"x=";MOUSE(1);"y=";MOUSE(2) GOTO
loop
```

Współrzędne ukażą się w okolicy lewego górnego rogu okna. Pomyśl, co należy zrobić, aby ukazywały się gdzie indziej.

I to już wszystko. Za miesiąc rozwiązania ćwiczeń z drugiej części kursu. Zachęcam do ciągłego eksperymentowania. Piszcie do redakcji o wszystkich swoich problemach (umieszczajcie na kopercie dopisek "kurs grafiki").

AMIGA PLAY

JA W KWESTII FORMALNEJ...

Szanowni Czytelnicy!

Począwszy od dwóch ostatnich wydań naszego pisma staramy się dostarczać Wam opisy i podpowiedzi do gier. Jednak dopiero teraz wyglądają one tak, jak powinny, gdyż wprowadziliśmy system oceny. Nie jest on taki sam, jak w naszym bratnim piśmie — AMIGA Magazin. Po dyskusji w gronie redakcyjnym doszliśmy do wniosku, że nie wszystko można bezkrytycznie przenieść z niemieckiego oryginału do jego polskiej edycji. Postanowiliśmy wprowadzić własny system oceny gier, może nie tak wymyślny, jak w AMIGA Magazin, jednak, naszym zdaniem, bardziej przystępny. Zastosowaliśmy oceny z przedziału od 1 do 9, ponadto 0 oznacza, że coś nie występuje (dla rzeczy tragicznych zarezerwowana jest jedynka) — 10 to gra daleko wybiegająca naprzód lub naprawdę świetna. Ta ostatnia nota może się niekiedy zdarzać w ocenach składowych, jednak gier, które dostaną jako ocenę ogólną — 10, będzie, według nas, nie więcej niż jedna czy dwie, w całym roku. Ocenę ogólną 9 dajemy grom, które szczególnie polecamy, wybór gier z niższą oceną zostawiamy już Czytelnikom.

Ocenie poddajemy cztery aspekty gry. Pierwszy to grafika — jak wiadomo, dobra gra powinna mieć dobrą grafikę. Oceniać będziemy nie tylko cechy artystyczne, ale także użytkowe — co z tego, że grafika prezentuje najwyższy poziom, jeżeli bardziej przyciąga uwagę niż akcja gry. Jest to ważne, gdyż zdarzają się programy (na razie nie podajemy nazw), w których niektórzy ludzie mają wielkie trudności z grą właśnie z powodu "przedobrzonych" grafiki...

Następną rzeczą, którą oceniamy, jest dźwięk. Jak ogólnie wiadomo, Amiga ma najlepsze możliwości muzyczne ze wszystkich obecnie popularnych na polskim rynku kompu-

terów. Oczywiście mowa o wersji podstawowej, gdyż na przykład użytkownik PC może włożyć do komputera kartę Roland LAPC, która w zasadzie jest przynajmniej półprofesjonalnym syntezatorem, lub popularnego Soundblastera i wmawiać nam, użytkownikom Amig, że jego PC lepiej gra od Amigi. Rzeczywiście gra lepiej — szczególnie w wersji podstawowej z pc-speakerem... Dość dygresji, powróćmy do głównego tematu. Otóż przez pojęcie "dźwięk" rozumiemy zarówno muzykę tytułową w grze, przeróżne "jingle", jak i efekty dźwiękowe (special fx). Tu również będziemy zwracać uwagę na ich funkcjonalność oraz realizm — wszak Amiga może wydobyć z siebie dowolny dźwięk.

Następną kategorią oceny jest pomysł. Przez to określenie rozumiemy tu oryginalność gry, odmiennosc od innych. Wykładnikiem jest tu środek skali ocen — jeżeli grze za pomysł przyznajemy 5, oznacza to, że nie wyróżnia się niczym szczególnym od innych. Oceny wyższe oznaczają, że gra zawiera coś ciekawego, nie spotykanego w innych jej kategoriach, natomiast interpretację ocen poniżej piątki zostawiamy Czytelnikom...

Przedostatnią oceną jest atrakcyjność. Zapewne tu będą największe kontrowersje — ocena ta zależy wyłącznie od upodobań osoby oceniającej. Oznacza to zapewne, że recenzent nie lubiący gier strategicznych oceni atrakcyjność o wiele niżej niż ich zwolennik. Jednak za punktację gry odpowiada wyłącznie recenzent i ma on prawo do ustalenia jej w sposób dowolny. Wszak nieraz się zdarza — biorąc przykład z innych gazet — że jedna i ta sama książka jest zupełnie niedoceniona w jednym piśmie, a w drugim dostaje tytuł "książki miesiąca". Na dodatek, nikt nie ma o to pretensji. My również liczymy na wyrozumiałość naszych Czytelników.

Ostatnią oceną jest ocena nazwana "ogólnie". Nie jest to wbrew pozorom średnia arytmetyczna poprzednich ocen, lecz

wrażenie całościowe, jakie gra wywarła na oceniającym.

Poniżej będziemy zamieszczać nazwę firmy wydającej grę — zwykle zagranicznej. W przypadku gier dostępnych na naszym rynku w oryginalnych kopiach podawać będziemy także polskiego dystrybutora, jeżeli oczywiście udostępnił nam dany tytuł. Nie jest to żadna forma kryptoreklamy, wybór gier do oceny nie zależy od dystrybutora i zastrzegamy sobie prawo swobodnego wyboru. W ekstremalnym przypadku może się zdarzyć tak, że daną grę ocenimy bez podania jej polskiego dystrybutora i oznaczać to będzie, że nie została nam przez niego dostarczona.

Przy ocenianiu gier liczymy także na współpracę Czytelników. Przesyłajcie nam własne recenzje, jeżeli uznamy, że nadają się do publikacji, to zamieścimy je, a autor dostanie tzw. wierszówkę zgodną ze stawkami obowiązującymi w naszym piśmie. Najlepiej teksty przysyłać na dyskietkach 3,5" sformatowanych na Amidze (880 KB) lub PC, (720 KB, 1,44 MB), które oczywiście zwrócimy. Prosimy o używanie polskich liter w standardzie Mazovii. Nie przysyłajcie nam tekstów o objętości mniejszej niż półtora strony maszynopisu, czyli 2500 — 2700 bajtów, jednak, aby zapobiec wodolejstwu czy "bajkotwórczości", zastrzegamy sobie możliwość skrótów. Przykładowe ekrany z gier można również przysłać do redakcji w formie standardowego obrazka IFF ILBM, jednak bez żadnych zmian — sprawdzimy to! Sposób

"ściągnięcia" ich z gier zostawiamy już inwencji Czytelników — polecamy sprzętowe przystawki, jednak może się zdarzyć, że ekran nie da się zgrać na dysk w ogóle lub nie uda się go zgrać w całości — wtedy nie należy przysyłać części obrazka — my się tym zajmujemy. Swoje produkcje przysyłajcie na adres naszego pisma Z WYRAŹNYM DOPISKIEM "AMIGA — RECENZJA" na kopercie. Przy okazji, nie odpowiadamy za pracę poczty i ewentualne trudności wynikłe z jej powodu.

Ostatnią rzeczą, którą chciałbym poruszyć, jest lista przebojów gier. Od tego numeru zaczynamy ją prowadzić. Oczywiście nie będzie ona miała sensu bez odzewu Czytelników, a więc prosimy o nadsyłanie WYŁĄCZNIE KARTEK POCZTOWYCH z głosami na 5 tytułów gier. Na kartce pocztowej wyraźnie dopiszcie "AMIGA — TOP20". Za udział w głosowaniu zapewne będą nagrody, więcej szczegółów przy samej liście. Jednocześnie wprowadzamy nowość na rynku: listę przebojów programów polskich. Pierwsze notowanie w następnym numerze. Zasady — jak wyżej, głosujemy na 5 pozycji, a na kartce pocztowej dopisujemy "AMIGA — POL5".

To na razie tyle w ramach wstępu do działu gier, spotkamy się za miesiąc, Czytelnicy liczą na ciekawe gry i odpowiedzi do nich, my liczymy na odzew od Czytelników. Do zobaczenia w numerze 3/92!

Wasz Rafał Włosna

MEGATRAVELLER II

300000 lat temu rozegrała się we wszechświecie straszna wojna. Obie strony konfliktu wyniszczyły się wzajemnie, używając apokaliptycznych broni. Jedynym świadkiem tamtych wydarzeń pozostały tajemnicze budowle, których przeznaczenia nie udało się współczesnym naukowcom wyjaśnić. Właśnie w jednym z takich miejsc, na planecie



Rhylanor, dochodzi do wypadku podczas trzęsienia ziemi. Zostaje uruchomione tajemnicze urządzenie. Skutkiem jego działania jest powstanie szybko rozprzestrzeniającego się śluzu o toksycznym działaniu. Planeta zostaje objęta kwarantanną, ludność częściowo ewakuowana. Naukowcy są bezsilni, gdyż śluz całkowicie odciął dostęp do biologicznej maszyny. Rhylanor to międzygalaktyczne centrum adminis-

tracyjno-gospodarcze, w którym krzyżują się interesy wielu ludzi i potężnych korporacji. Czy wydarzenia w tajemniczej budowlu to tylko zbieg okoliczności, czy też mieli w nich udział dwaj tajemniczy osobnicy, widziani przez świadków?



Twoim zadaniem jest wyjaśnić tę zagadkę. Jako nagrodę za uratowanie planety bądź części jej powierzchni, rząd i popierające go korporacje przeznaczyły ogromną kwotę — pół miliarda kredytów. Sam jednak nie podążasz temu trudnemu zadaniu, możesz sobie dobrać czterech współtowarzyszy (przebywających do tej pory w miejscowym barze). Wybierając ich rasę, płeć, świat, z którego pochodzą, oraz zawód (spośród kilkunastu możliwych, od piratów poprzez oficerów armii i kupców aż do naukowców) determinujesz ich cechy i zdolności. Możesz je modyfikować przez dobór typu szkolenia. Powinieneś zwrócić bacniejszą uwagę na warunki fizy-

czne każdej postaci bądź też skoncentrować się na doskonaleniu jej umiejętności w zakresie obsługi urządzeń i pojazdów. Nie należy zapominać także o tym, że Twój zespół będzie narażony na liczne niebezpieczeństwa, tak więc sztuka posługiwania się różnorodną bronią nie powinna być mu obca. Program pozwala Ci skorzystać z gotowych już postaci, możesz je też generować sam.



Gra jest dość rozbudowana. Akcja toczy się na różnych planetach, musisz spenetrować wiele tajemniczych miejsc, nawiązywać kontakty z napotkanymi osobami, staczać walki zarówno na Ziemi, jak i w kosmosie. Program dostarcza wielu możliwości zdobywania funduszy w twojej wędrowce przez wszechświat. Można trudnić się galaktycznym handlem,



spieniężać znalezione cenne przedmioty lub też zabawić się w łowcę nagród za głowy przestępców! W każdym porcie kosmicznym znajdziesz: sklepy, komendę policji, szpital, biura podróży turystycznych, ośrodki szkolenia armii lub floty i wiele innych. Napotkane osoby będą wykazywały różny stopień uprzejmości w stosunku do Ciebie. Jedne bez wahania udzielą Ci informacji, inne zrobią to dopiero po otrzymaniu sowitej łapówki. Dbaj więc o stan swoich funduszy.

Sterowanie zespołem odbywa się zarówno za pomocą myszy, jak i klawiatury. Wybór dostępnych opcji jest dość przejrzysty i nie nastręcza większych problemów w posługiwaniu się (dobrze dobrany zestaw ikon). Grafika nie oszałamia, szczególnie widok z góry przy poruszaniu się po

wierzchni planety, ale dzięki swej różnorodności i wpleceniu wielu sekwencji animowanych jest do przyjęcia. Jako podkład muzyczny możemy wybrać muzykę ze wstępu (dobrze dobraną nastrojowo) bądź też efekty dźwiękowe, również na niezłym poziomie. Produkt firmy Paragon Software zajmuje trzy dyskietki i wskazane jest posiadanie drugiej stacji dysków, w przeciwnym razie należy się przygotować na uciążliwą żonglerkę. Program Megatraveller II polecam przede wszystkim fanom gier typu adventure lub role-playing, którzy mają dużo wolnego czasu. Znajdą w nim oprócz standardowych elementów wiele nowych, ciekawych pomysłów.

Roman Sadowski

MEGATRAVELLER II		Liczba
Paragon Software		
Grafika	★★★★★☆☆☆☆	6
Dźwięk	★★★★★☆☆☆☆	6
Pomysł	★★★★★☆☆☆☆	8
Atrakcyjność	★★★★★☆☆☆☆	7
Ogólnie	★★★★★☆☆☆☆	7

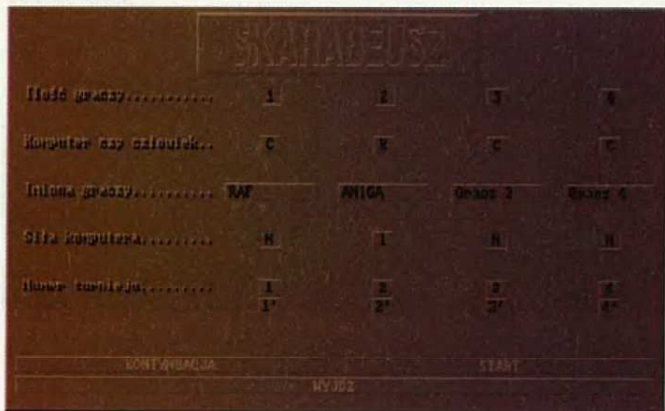


SKARABEUSZ

SCRABBLE jest grą szeroko rozpowszechnioną na świecie. U nas w Polsce rozpopularyzowana została na łamach niezapomnianego tygodnika "Razem" — organizuje się nawet mistrzostwa polski w SŁÓWKA — bo tak brzmi polska nazwa tej gry. Zasady uczestniczenia w niej są bardzo proste: na 225-polowej, różnokolorowej planszy (15 na 15 pól) układamy wyrazy z klocków oznaczonych literami. Gracze mają po siedem klocków i mogą używać wyłącznie tych, które właśnie trzymają w dłoni. Każda litera ma swoją wartość wypisaną w prawym dolnym rogu klocka. Za ułożenie wyrazu uzyskujemy tyle punktów, ile jest w sumie na wszystkich jego literach. Różne kolory pól planszy oznaczają specjalne właściwości. Jeżeli klocek ustawimy na polu jasnoczerwonym, to suma punktów wszystkich liter podwaja się, podwaja się także wartość litery ustawionej na polu jasnoniebieskim. Podobnie

jest z polami ciemnoczerwonymi i ciemnoniebieskimi — tutaj jednak zachodzi potrojenie.

Wyrazy, które układamy, muszą — rzecz jasna — coś znaczyć. Mogą to więc być rzeczowniki w obu liczbach, przymiotniki w mianowniku obu liczb w dowolnym rodzaju, czasowniki w dowolnej osobie obu liczb w czasie teraźniejszym w trybie orzekającym, zaimki lub nazwy własne. Ponadto, wyrazy należy "przykładać" poziomo lub pionowo do istniejących już liter, inaczej ruch jest nieważny. Powstała konstrukcja ma przypominać krzyżówkę.



W czasie gry możliwa jest wymiana klocków w dłoni. Pozwala to pozbyć się nieużytecznych, choć zwykle dużo wartych liter, zamieniając je na losowo wybrane z pozostałych. Niekiedy zdarzy się, że dostaniemy literę pustą, zaznaczoną jako znak podkreślenia ("_"). Jest ona warta 0 punktów, ale za to może zastąpić dowolną inną literę wybraną przez gracza. I to właściwie tyle zasad gry — reszta należy do grających. Warto jedynie wspomnieć, że w przypadku wykorzystania, do ułożenia wyrazu na planszy, wszystkich siedmiu liter z ręki dostajemy ekstra 50 punktów.



Zasady gry są wyjątkowo proste, jednak wymagają od graczy dobrej znajomości języka oraz dużego zasobu słów. Mimo to, jest to wręcz wymarzona gra do implementacji na komputery. Na Amigę powstało kilka wersji SCRABBLE, lepszych lub gorszych, jednak nasi rodzimi programiści słusznie stwierdzili, że są one wyłącznie angielskojęzyczne i przydałaby się wersja polska. I nie mylili się — gra w SKARABEUSZA,

polskie wydanie SCRABBLE, jest bardzo atrakcyjna. Tu mała dygresja. Otóż daje się zauważyć, że świeżo upieczony programista zaczyna od gier, na które zawsze znajdują się chętni — zwykle są to gry logiczne. Przykładem mogą być członkowie grupy Crusaders, których pierwsza gra była dwuwymiarową wersją kostki Rubika. Podobnie jest z innymi i można wysnuć wniosek, że pisanie takich gier jest prostsze od napisania na przykład Battle Squadron czy Lemmingów. Programy takie wymagają bardzo długiego czasu włożonego w przygotowanie systemu, na przykład wyświetlania obiektów na ekranie, czasu na zaprojektowanie grafiki tła i ułożenie jej w przesuwaną się krajobraz (lub w plansze, jak to ma miejsce w Lemmingach). Jedna osoba obciążona wszystkimi tymi obowiązkami (oczywiście wyłącznie od strony programowania; nie można wymagać, żeby programista był zarazem muzykiem i grafikiem w jednej osobie) jest bardzo nieefektywna i stworzenie gry jest o wiele kosztowniejsze niż przy pracy zespołowej. A gry logiczne wcale nie jest tak trudno zaprogramować, na pewno łatwiej niż "rasowy" produkt i dlatego też zwykle od tego się zaczyna. Dość jednak, wróćmy do SKARABEUSZA.



Jest to, o ile pamiętam, pierwsza polska gra, a jeżeli nie, to pierwsza tak rozprowadzana. Napisał ją pan Marek Hyla z Krakowa, jedna z bardziej znanych postaci w świecie Amigi, przynajmniej z czasów ACC. Jest on pionierem rodzimego oprogramowania, jako jeden z pierwszych napisał słownik angielsko-polski. Teraz jego SKARABEUSZ może stać się wizytówką autora.

Program jest napisany prawdopodobnie w języku C, można to wywnioskować z charakterystycznej dla tego języka powolności. Najpierw jesteśmy raczeni gustownym ekranem początkowym, a potem planszą, na której możemy wybrać liczbę graczy (nie ilość, jak to jest napisane — gracze są niewątpliwie policzalni), którymi sterować będzie komputer (można zmusić program do zagrania partii, w której wszystkimi czterema graczami steruje maszyna), nazwy uczestników, siłę komputera (która jest proporcjonalna do czasu potrzebnego na wykonanie ruchu przez Amigę). Ponadto program pozwala rozegrać do czterech partii w tym samym gronie graczy. Punktacje oraz wszystkie dane turnieju można nagrać na dysk. Mamy jeszcze możliwość przywołania gry z dysku, wyjścia z programu oraz rozpoczęcia rozgrywki. Przedostatnia opcja jest, według mnie, źle pomyślana — powoduje wyzerowanie komputera zamiast wyjścia z powrotem do CLI, z którego program został uruchomiony...

Grę rozpoczyna wybrany losowo gracz. Jeżeli jest to człowiek, to musi ustawić pierwszy wyraz tak, aby jedna z jego liter leżała w centralnym miejscu planszy. Inni gracze dokładają już swoje litery do leżących na planszy. Gdy gracza zastępuje komputer, w miejscu ikon gry (w prawej, górnej części ekranu) znajdują się "oczy" komputera. Co pewien czas zamykają się — pozwala to stwierdzić, że komputer próbuje wybrać najlepszy ruch. Czas oczekiwania na odpowiedź programu jest niekiedy niezwykle długi, nawet gdy ustawimy siłę komputera na 1.

Litery na klockach w ręku gracza, który ma właśnie wykonać ruch, są w kolorze różowym. Jeżeli graczem jest człowiek, program czeka na jego posunięcia. Myszka wskazuje pole,



gdzie chcemy wstawić literę, mlaskamy lewy jej przycisk, a następnie mlaskamy nad literą z ręki lub wpisujemy ją z klawiatury (przepraszam za "bieleckość" tekstu). Tu czeka na nas niespodzianka — otóż litery Z i Y są umieszczone jak w klawiaturze niemieckiej, co może sprawić pewne kłopoty ludziom przyzwyczajonym do układu angielskiego (a zapewne jest ich większość). W przypadku gdy stwierdzimy, że coś jest nie tak, można jednym przyciśnięciem myszki nad odpowiednią ikoną "zdjąć" wszystkie klocki z literami z planszy. Można też wymienić litery w dłoni na losowo wybrane z reszty, która pozostała (ile ich jest, widać w prawym dolnym rogu — nie można tego zrobić, kiedy wskaźnik jest czerwony), jednak kosztować będzie to nas ruch. W ekstremalnym przypadku możemy opuścić ruch, a gdy zrobią to wszyscy gracze, rozgrywka zostaje zakończona. Wtedy od wyniku uzyskanego przez danego gracza jest odejmowana suma liter, które pozostały w ręce, a osoba (maszyna?), która po tej operacji będzie miała najwięcej punktów, wygrywa. Początek słownika SKARABEUSZA składa się wyłącznie z rzeczowników, jednak program uczy się wyrazów nie znanych mu, których użył człowiek. Można je potem zapisać i na stałe włączyć do zasobu słów gry.

Całość oprawiona może być w dość średnie efekty dźwiękowe lub lepszą od nich muzykę — możliwe jest przełączenie w czasie gry. Grafika nie prezentuje najwyższego poziomu — mogłaby być lepsza — jednak spełnia swoje zadanie. Obsługa gry nie jest trudna, poza tym do dyskietki dostarczana jest instrukcja obsługi — 12-stronicowa broszurka. W celu ochrony przed nieuczci-

wymi użytkownikami program jest indywidualnie oznakowany, a poza tym naprawdę bardzo sprytnie zabezpieczony — nie powiem jak, jednak mogę stwierdzić, że takie zabezpieczenie jest bardzo skuteczne, o wiele bardziej niż przemysłowe systemy zapisu specjalnych danych, na przykład na ścieżce 82...

Całość sprawia dobre wrażenie, bardzo przyjemna jest gra w cztery osoby i zdecydowanie polecam SKARABEUSZA wszystkim. Łączy on przyjemne z pożytecznym — wszak gra w niego jest świetnym ćwiczeniem z polskiego! Należy tu jednak zaznaczyć, że nie jest najlepsza dla uczniów wczesnych klas szkoły podstawowej — po prostu nie mają oni tak dużego zasobu słów, jakiego gra wymaga. Niemniej, program wspaniale nadaje się dla szkół wyposażonych w Amigę i może być ciekawszym zajęciem niż, na przykład, uczenie się wierszy na pamięć...

Rafał Wiosna

SKARABEUSZ		Liczba
Twin Spark Soft		
Grafika	★★★★★☆☆☆☆	6
Dźwięk	★★★★★☆☆☆☆	6
Pomysł	★★★★★☆☆☆☆	9
Atrakcyjność	★★★★★☆☆☆☆	9
Ogólnie	★★★★★☆☆☆☆	8

LURE OF TEMPTRESS

Roman Sadowski

W kraju rządzonym przez mądrego króla zapanał spokój. Ucichły spory między mieszkańcami, a od wroga zewnętrznego skutecznie chroniło morze i nieprzebyte góry. Ludność zajęta się zbiorami, zaś monarcha mógł oddać się swej ulubionej rozrywce — polowaniu. Idyllę tę przerwał posłaniec przynoszący niepokojące wieści. Następnego dnia,



nie opodal miasta Tornvale władca i jego mała eskorta stoczyli nierówny bój z armią bestii, które niczym zjawy wyłoniły się z porannej mgły. Dowodziła nimi piękna, a zarazem przewrotna, czarodziejka Selena. Bitwę przeżyła tylko jedna osoba — przypadkowy świadek Diermot. Ogluszony po upadku z konia, został odnaleziony przez zwycięskiego wroga i wtrącony do lochów Tornvale.



I w tym miejscu rozpoczyna się akcja gry "Lure of Temptress" napisanej przez Revolution Software Ltd. Moim skromnym zdaniem program ten ma szansę stać się szlagierem w klasie gier przygodowych. Pierwsze, co się rzuca w oczy, to dopracowana grafika. Duża liczba detali, płynna animacja to tylko niektóre jej zalety. Oprawa muzyczna prezentuje również wysoki poziom. Autorzy pokusili się o pełne odwzorowanie



odgłosów rzeczywistego świata (na przykład trzask ognia w kominku, szczęk kufli w barze itd). Zupełnie nowym pomysłem jest obdarzenie występujących w grze postaci pełną inteligencją. Prowadzą one niezależne życie, rozmawiają ze sobą, załatwiają własne sprawy. Ty jako główny bohater możesz stać z boku i obserwować je bądź też bezpośrednio ingerować w ich działania. Każda osoba ma specyficzny charakter, w różny sposób reaguje na Twoje wysiłki zmierzające do rozwikłania tajemnic. Często będziesz miał do wyboru podczas dialogów wiele zwrotów o różnym znaczeniu. Od ich prawidłowego wyboru, którego dokonasz na podstawie zgromadzonych już informacji, będzie zależał Twój dalszy los.

Niektóre fragmenty gry wymagają szybkich działań, nie możesz pozwolić sobie na chwilę zawahania, w przeciwnym razie spotka Cię rychła śmierć z ręki Skorla. W Twojej wędrówce po Tornvale przyjdzie Ci również prowadzić walki. Są one dość ciekawie rozwiązane. Całe sterowanie odbywa się wtedy za pomocą myszy. Od tego, na jaką część Diermota wskażesz, zależy rodzaj zadawanego ciosu, zaś o tym, jaki ma on mieć charakter — ofensywny czy obronny, decyduje to, który przycisk myszy naciśniesz. Niezwykle ważne jest dokładne oglądanie każdego pomieszczenia, często pewne elementy przedmiotów ujawniają się po ich dokładnym obejrzeniu. Na przykład zamek w drzwiach stanie się widoczny dla



Twojego bohatera dopiero po wydaniu mu polecenia "look at door". W pewnych sytuacjach nie będziesz w stanie poradzić sobie z wykonaniem niektórych czynności, ale od czego masz przyjaciół? Możesz zwracać się do nich o pomoc, przy czym Twoje polecenia mogą być bardzo skomplikowane i zawierać do kilku komend. Program charakteryzuje się olbrzymią dbałością o zgodność z rzeczywistym światem, wszystko ma swój logiczny sens. Osoba, która weszła do pomieszczenia, nagle nie znika, pozostaje tam (możesz sprawdzić, czym się zajmuje, podglądając ją przez okno).



Autorzy twierdzą wręcz, że stworzyli nowy gatunek gry nazwany przez nich "Virtual Theatre". W programie do perfekcji dopracowany jest sposób komunikacji z otaczającym Cię światem. Przypomina on system użyty w takich grach jak

Kult czy Operation Stealth. Wszelkie komendy wydawane są za pomocą myszy, poprzez wybór odpowiednich opcji. Twoim zadaniem jest jedynie najechanie na wybrany przedmiot bądź osobę i kliknięcie (lewy przycisk służy do oglądania rzeczy, zaś prawy daje dostęp do czynności, jakie możesz w danym momencie wykonać). W miarę postępów w grze uaktywniają się kolejne dostępne polecenia, a napotkane osoby mogą inaczej niż przedtem odnosić się do Twoich działań. Program zajmuje cztery dyskietki, współpracuje z dwoma stacjami dysków i działa na 1 MB (nieodzowne jest zarezerwowanie piątej dyskietki na zapisy). "Lure of Temptress" można z czystym sumieniem polecić każdemu — nie tylko zaawansowanym graczom, ale także tym początkującym, którzy tylko trochę znają język angielski.

LURE OF TEMPTRESS		Liczba
Virgin Game		
Grafika	★★★★★★★★	9
Dźwięk	★★★★★★	7
Pomysł	★★★★★★	7
Atrakcyjność	★★★★★★	8
Ogólnie	★★★★★★	8

ZONE WARRIOR

Mamy rok 2967. Od ponad dziesięciu lat wrogie Imperium Geeków próbuje podbić Zjednoczenie Ludzi i Innych Form Życia na Ziemi na wszelkie sposoby. Teraz przygotowują swój decydujący i najgroźniejszy atak, ze wszystkich, których do tej pory się dopuścili. Dzięki ich maszynie czasu zamierzają się przenieść w przeszłość, aby zniszczyć Ziemię. Czas nagli. Najsprawniejszy z żołnierzy Zjednoczenia musi szybko wybrać się w przeszłość, aby obronić Ziemię i Inne Formy Życia. Tym żołnierzem jesteś Ty. Twój pseudonim: Zone Warrior.



"Standard" — pomyśli zapewne część Czytelników. Historyjka prawie jak z brukowej powieści science-fiction lub z... gry komputerowej. I dokładnie tak jest — tak przedstawia się fabuła gry Zone Warrior, wydanej przez znaną firmę Electronic Arts. Jak można się spodziewać, jest to gra typu bij-zabij,

gdzie myślenie nie jest zwykle wymagane, za to refleks jest jedną z podstawowych cech, którą musi mieć grający. Od dawna takie gry cieszyły się popularnością, szczególnie było to widać na komputerkach 8-bitowych, gdzie w takie hity jak Terra Cresta, Xevious czy Commando grało się godzinami. Również na 16-bitówce napisano wiele gier takiego typu i cieszą się one powodzeniem.



Po wczytaniu gry raczeni jesteśmy sekwencją wprowadzającą, z której można się dowiedzieć, z jakiego powodu i po co wybieramy się w przeszłość. Trzeba przyznać, że część ta nie jest najgorsza, choć widziano lepsze. Następnie, po kilkukrotnej zmianie dysków, możemy zagrać. Idea jest prosta — strzelaj do wszystkiego, co się rusza. Tym razem poruszamy główną postacią, którą jest uzbrojony po zęby żołnierz z budzącą szacunek łufą na ramieniu. Możemy poruszać się w lewo, prawo oraz wykonywać skoki. W przemieszczaniu się przeszkadzają przeróżne stwory — w pierwszej fazie gry są to



prehisteryczne gady, jako że przeniesieni zostaliśmy do epoki, w której czują się one najlepiej — epoki kamienia łupanego. Jak można się domyślić, każde zetknięcie się z nimi powoduje utratę sił oraz życia.

Swoją drogą to ciekawe, wnioskując z gier komputerowych zjawisko reinkarnacji jest jedną z wrodzonych cech człowieka, często wykorzystywanych wszędzie, gdzie tylko można dostać przysłowiową "kulkę w łeb". Szkoda jednak, że nie można tak naprawdę...

Wracając jednak do gry, w miarę jej upływu przenosisz się w czasie, oczywiście, jeżeli jesteś w stanie stawić czoła wszystkim przeciwnościom z tym związanym. Twoje rozkazy są proste: 1. MUSISZ wykonać powierzoną Ci misję, nie ma innego wyjścia. 2. Staraj się nie dać zabić — pamiętaj, że martwy żołnierz nie jest przydatny w działaniach wojennych. 3. Ratuj więźniów (są to zwykle długonogie blondynki...). 4. Uwolnij Boda, najważniejszego więźnia, bez którego nie jest możliwe zatrzymanie Imperium Geeków. 5. Zabij ich przywódcę (jak zwykle...). Przy okazji możesz dowolnie zabijać szeregowych Geeków, jak i inne stwory, które mogą Ci zagrozić.

W miarę poznawania krainy możesz znaleźć bardzo przydatne rzeczy — jak na przykład przystawkę do twojej lufy powodującą wzrost siły rażenia, substancje przyspieszające poruszanie się lub mapy terenu. Są to niezbędne rzeczy do ukończenia misji, więc staraj się ich jak najwięcej znaleźć.

Gra jest typową "strzelanką" wykonaną na średnim poziomie — nie jest to coś specjalnego, ale też trzyma pewien standard wytyczony przez komputery 16-bitowe. Program sprawia wrażenie bezpośrednio przeniesionego z Atari ST, gdyż animacja jest typowa dla tego komputera (czyt.: nie jest płynna, fachowcy

mówią "50 ramek na sekundę") i kolory są dość ciemne (każda składowa koloru na Amidze to 16 wartości, na ST — tylko 8, więc gdy bezpośrednio przenosi się dane graficzne z ST na Amigę, obraz zwykle jest o połowę ciemniejszy). Zone Warrior dedykowany jest ludziom, którzy lubią ten typ gier — są wszak tacy, którzy potrafią przejść "Battle Squadron" czy "Ikari Warriors" nie tracąc ani jednego życia. Jediną zaletą takich gier jest to, że są tanie i dzięki temu mogą się cieszyć powodzeniem wśród nabywców.

Rafał Włosna

ZONE WARRIOR		Liczba
Electronics Arts		
Grafika	★★★★★☆☆☆☆	6
Dźwięk	★★★★★☆☆☆☆	7
Pomysł	★★★☆☆☆☆☆☆	3
Atrakcyjność	★★★★★☆☆☆☆	5
Ogólnie	★★★★★☆☆☆☆	5

NIELEGALNY DOPING

cz. 2

W tym wydaniu znowu sięgamy do tajemnej księgi gier. Najpierw jednak zaległość z numeru pierwszego:

Armalyte

W trakcie gry wciśnij pauzę i wpisz DELTA 3, a zostaniesz nieśmiertelny.

oraz zaległość z numeru zerowego:

International Karate +

Po udanym ciosie przeciwnika naciśnij klawisz spacji, a następnie przycisk FIRE.

Dalej prezentujemy "tyłne wejścia" do gier od litery C do L. Przy okazji zachęcamy Czytelników do przysyłania nam swoich odkryć na tym polu, każdą poprawkę na pewno obejrzymy i jeżeli będzie nadawała się do publikacji (tzn. nie będzie się powtarzała), to zamieścimy z podaniem imienia i nazwiska (lub pseudonimu) osoby, która ją przysłała. Czekamy na listy przysyłane na adres redakcji z dopiskiem "GRY-NIELEGALNY DOPING".

Cybernoid

Na ekranie tytułowym wpisz RAISTLIN i naciśnij spację. Wciśnięcie N zmienia aktualny poziom. Możesz też

spróbować zdefiniować klawisze sterowania jako Y X E S (w takiej kolejności), aby uzyskać nieskończenie wiele statków.

Cybernoid II

Wpisz NECRONOMICON na ekranie tytułowym. Podobnie jak w pierwszej części gry wciśnięcie N zmienia poziomy. Również zdefiniowanie klawiszy sterowania jako Y G R O da Ci nieskończenie wiele statków.

Darkside

Przytrzymaj klawisze 2 i 8 i naciśnij przycisk FIRE — zobaczysz zdjęcia programistów.

Elite

Kilka nie udokumentowanych możliwości gry: gdy ujrzysz ekran tytułowy z obracającym się statkiem, wciśnij A — to spowoduje, że przestanie on się kręcić. Teraz możesz poruszać nim za pomocą klawiszy kursora lub A+klawisz kursora (szybciej). Istnieje możliwość przybliżania i oddalania statku poprzez wciśnięcie O lub I. Jeżeli zabawa Ci się znudzi, wciśnij D. W czasie lotu naciśnij W, aby zapoznać się ze swoimi wynikami oraz z tym, kto zrobił grę. Wpisz SARA jako hasło i naciśnij RETURN. Wejście do zastrzeżonego ekranu jest możliwe przez naciśnięcie "gwiazdki".

Exolon

Wpisz w tablicę wyników AD ASTRA, aby uzyskać nie-
skończone życie.

F-29 Retaliator

Wpisz się jako DIDYMEN, kliknij ikonę COLONEL i wciśnij
RETURN.

Flood

Oto kody do pierwszych 42. poziomów:

01 - FROG	08 - GRIP	15 - FOUR	22 - REED	29 - LOOP	36 - BRIL
02 - YEAR	09 - TRAP	16 - GRIT	23 - LIME	30 - SING	37 - EGGS
03 - QUIF	10 - THUD	17 - ZING	24 - QUID	31 - JOUX	38 - HENS
04 - LONG	11 - FRAK	18 - JING	25 - WING	32 - PINK	39 - HAIL
05 - WORD	12 - VINE	19 - LIDO	26 - FLEE	33 - GOGO	40 - SOAP
06 - FRED	13 - UMP	20 - POOL	27 - GIGA	34 - LETS	41 - FOAM
07 - WINE	14 - NILL	21 - HATE	28 - HEAD	35 - QUAD	42 - MEEK

Ghostbusters II

Przytrzymaj klawisze ALT CTRL 5 U, gdy na ekranie znaj-
duje się znaczek firmy Activision i wciśnij przycisk FIRE.

Ghost'n'Goblins

Wpisz na obrazku z informacjami o autorach gry wyraz
DELBOY, a otrzymasz nieśmiertelność.

Ghouls'n'Ghosts

Tuż po rozpoczęciu gry wpisz KAREN BROADHURST, a
uzyskasz nieśmiertelność.

The Great Giana Sisters

Wpisz ARMIN, aby przenieść się o jeden poziom wyżej.

Hard drivin'

Spróbuj uzyskać maksymalną szybkość i nacisnąć N.
Ułatwi Ci to sterowanie oraz da nieśmiertelność i nieskończo-
ny czas.

Hollywood Poker Pro

Gra dla prawdziwych mężczyzn. Po wybraniu partnerki do
gry i ukazaniu się jej obrazka z lewej strony na-
cisnij SHIFT + H + F9 i nie puszczać tych klawiszy, nacisnij
trzecią ręką dwa razy przycisk DROP na ekranie. Twoja przeci-
wniczka posłusznie zrzuci część garderoby. Gdy za pierwszym
razem się to uda, możemy całą sytuację powtórzyć, jednak w
tym przypadku DROP wciskamy tylko raz. Ponadto, gdy przeci-
wniczka zacznie obrzucać Cię wyzwiskami, wciśnij klawisz A i
napisz jej coś "od serca", oczywiście w języku angielskim.

Hybris

Jest to poprzednik Battle Squadron. Na ekranie
najlepszych wyników wpisz COMMANDER i wciśnij przycisk
FIRE w joysticku. Następnie wciśnij F10. Uzyska-
łeś nieśmiertelność — statek powinien drgać. Ponadto F1 do
F5 rozbudowuje statek, F8 zmienia poziom gry, a F10 wyłą-
cza/włącza tryb "szkoleniowy".

Interphase

W czasie gry wpisz Fenny (przez duże F). Po wciśnięciu
lewego przycisku myszy będziesz w stanie oglądać przedmioty.

Jumping Jackson

Oto hasła do poziomów: A — ROCK'N'ROLL, NOISES,
TENEBRE; B — SYNTHE, FUNK, ELVIS. Ponadto spróbuj
wciśnąć prawy przycisk myszy na ekranie tytułowym.

Last Ninja III

Oto kody, które możemy wpisać w tablicy najlepszych
wyników. Przeniosą nas one do poziomów: 1 — SUSS, 2 —
IMED, 3 — URTI, 4 — BASD, 5 — NOUS, 6 — REOO.

Led Storm

Gdy rozpoczniesz grę, wpisz: AMIGA DAVIDBROAD-
HURSTWANTSTOCHEAT (bez jakichkolwiek spacji). Da Ci to
nieśmiertelność.

Lemmings

Oto kody do poziomów:

Fun

1 -	11 OHODHCELCS	21 NNHCAROFDV
2 KJULDLCCCN	12 KMDLCINMCL	22 FLCMKLLGDJ
3 NJLDLCADCY	13 MDLCARLNCS	23 LCENLLFHDV
4 HNLHCIOECW	14 DLICIJNMOCM	24 CMNMLFLIDP
5 LDLCAJNFCK	15 LCAOLMDPCY	25 CCJMFLLJDDQ
6 DLICIJNLGCT	16 CMOLMDLQCV	26 IJMFLLCKDX
7 HCAONNLHCX	17 CEJHLFLBDX	27 NJMFLCCLDW
8 CINNLDLICJ	18 KKILFHCCDM	28 HMFHCIONDY
9 CEKHMDDLJCO	19 NJNNHCADDR	29 MNHCARKNNDL
10 IKHMDLCKCT	20 JLFLCKNEDW	30 FLCOKLMDU

Tricky

1 HCGOLONPDX	11 CEKKOLIJEK	21 OHLFMCADFN
2 CMOLMGLQDU	12 IKHMEMCKEK	22 ILFMCKOEFJ
3 CCJULDOBEJ	13 OJOLICLEN	23 LFMCEJMFFT
4 IKHLDCCEN	14 HMDCIOMEM	24 GICKKOLGFK
5 OHNLIBADEP	15 MDMCGJMNEL	25 MCGNMLFHL
6 IDLMCIOEEU	16 LMBIJOOOJ	26 BKONFIIFO
7 LDMCCJNFEP	17 KCCOMLPEU	27 CAJWGMJFS
8 EMCIJMLGEW	18 CRNLMDMQEV	28 KKKOICKFV
9 MCEOLLDHEW	19 CCKHNNKBFR	29 OHMFWCALFW
10 CMNLLEMIEP	20 IJULGMCCFS	30 HMGCMOMFU

Taxing

1	MFMCGKLNFO	11	LGANOLDHGJ	21	GGKHNNHBY
2	GMCOKLMOFY	12	GKNOLDLIGU	22	KJLFLNGCHK
3	MCANMFPFO	13	GAJIMMHJGX	23	NJLNGADHW
4	CINOMGMQFK	14	OKHMDNGKGT	24	KNOHGOEHL
5	GEKKNMJBGJ	15	OIMELGALGK	25	LFNGCJNFHJ
6	KJLLDLGCGU	16	HMDLGKMGU	26	GNGIULLGHP
7	NHNLJGCDGY	17	MELGCKLNGN	27	HFAMLLFHHU
8	ILDLMGOEGN	18	DLGIJOMOGV	28	FINLLFHHN
9	LDLGCJOFV	19	LGENMMDPGU	29	FAJHMFHJHP
10	ENGIUNLGGO	20	GKNNOLHOGS	30	IJHMFMPKHG

Mayhem

1	NJMNLFELHX	11	LDIFAJLFIO	21	IFANLMDPIN
2	HMFJFINMHQ	12	DIFIJLLGIX	22	FINLMDIQW
3	MFHFAJLHGX	13	MGCNNLEHIO	23	FAJHLFIBJJ
4	FHFIJLMOHQ	14	FINLLDIIN	24	IJHLFIFCJS
5	HFANLMPFHN	15	FAJHMDIIP	25	NHLFIFADJP
6	FINLMPQHW	16	IJHMEMGKIO	26	HLFIFINEJY
7	FAJHLDIBIW	17	NHMDIFALIV	27	LFIFAJLFR
8	IJHLDFICIP	18	HMDIFINMIO	28	FIFIJLLGJK
9	NHLDFIPADIN	19	MDIFAJLNIX	29	IFANLLPHKX
10	HLDFIFINEIV	20	DIFIJLMOIQ	30	FINLLFIIJQ

Na dzisiaj tyle, następna część w następnym numerze. Nie zapomnijcie! Czekamy na poprawki od Was!

Rafał Wiosna

PODPowiedzi DO CIVILIZATION

cz. 2

Kontynuujemy cykl podpowiedzi do gry CIVILIZATION, wydanej przez firmę Microprose.

PIERWSZE SPOTKANIE Z INNYMI CYWILIZACJAMI

1. Zawsze zgadzaj się na pierwszy oferowany pakt pokojowy przy pierwszym kontakcie z nieznaną Ci cywilizacją. Nie kosztuje to nic, a pozwala na przygotowanie się do bardziej antagonistycznych stosunków w razie potrzeby.

2. Po zawarciu paktu użyj oddziałów milicji (Militia), aby na bieżąco mieć wiadomości o rozwoju przeciwnika. Rozmieść oddziały w miejscach strategicznych i ufortyfikuj je, aby ustrzec się przed "podstępny" atakami z zewnątrz.

3. Jeżeli wystawiłeś oddziały graniczne, nie zapomnij o wsparciu ich innymi jednostkami na swoim terytorium, szczególnie gdy granicę broni milicja lub dyplomaci, których zdolność obrony jest ograniczona. Mocniejsze oddziały niedaleko granicy, lub mogące szybko przybyć w jej okolicę, będą bardzo pomocne i mogą zatrzymać atak.

4. Wyślij parę jednostek dyplomatów na terytorium potencjalnego przeciwnika tak szybko, jak to jest

możliwe po podpisaniu z nim paktu. Podczas jego obowiązywania dyplomaci i karawany mogą poruszać się bez przeszkód — to da Ci szansę wyrobienia wyobrażenia o potencjale twojego sąsiada, o jego silnych i słabych punktach.

5. Jeżeli spotkasz przeciwnika na morzu, popłyń za nim. Zwykle wrogowie odkrywają przed nami najkrótsze drogi morskie między lądami.

6. Wyślij karawany na terytorium przeciwnika, nawet jeżeli chcesz go zmieść z powierzchni planety. Zarabiaj, kiedy możesz!!

7. Użyj swoich statków, aby blokować — lub obserwować — porty przeciwnika. Jeżeli chcesz dominować nad światem, będzie Ci na rękę, jeżeli przeciwnik uplasuje się w jednym miejscu na mapie — nie będzie miał kolonii na innych kontynentach. Jeżeli jesteś nastawiony bardziej pokojowo, podążanie za statkami innych bander może dać Ci dobre mniemanie o tym, jakie są zamiary osadnicze innych.

8. Zwróć uwagę na naturalne bariery utrudniające możliwość ekspansji innych cywilizacji — mogą to być na przykład jeziora lub zatoki — i ustaw oddziały swojego wojska w strategicznych miejscach. Możesz też użyć oddziałów osadników (Settlers).

9. Jeżeli będzie Cię na to stać, możesz umieścić dyplomatów ufortyfikowanych lub nieaktywnych (Sentry unit) na terytorium przeciwnika. Da Ci to informacje o jego posunięciach militarnych.

DRUGIE MIASTO

1. Zbuduj swoje drugie miasto w najbardziej idealnym miejscu, jakie możesz znaleźć — czas nie gra już takiej wielkiej roli, jak w przypadku budowy pierwszego miasta.

2. Od razu zacznij tam budować koszary (Barracks) i spichlerz (Granary). Następnie ufortyfikuj w mieście kilka oddziałów, zdolnych je bronić.

3. Wyślij osadników z pierwszego miasta, aby rozwinęły otoczenie drugiego w czasie, kiedy jest zajęte produkcją rzeczy ważniejszych.

4. Jeżeli masz fundusze, możesz kupić budowle, które są niezbędne w początkowym okresie.

5. Przynajmniej jedno z dwóch pierwszych miast powinno być portem.

6. Zbuduj drogę między miastami tak szybko, jak to możliwe. Będą do tego potrzebni osadnicy.

7. Jeżeli wrogie cywilizacje leżą na zachodzie, zbuduj swoje miasto w poznanych przez Ciebie wschodnich rejonach, jeśli na wschodzie to odwrotnie. Pozwoli to uniknąć zaskoczenia w przypadku ataku.

8. Podobnie jak przy pierwszym mieście rozmieść oddziały obronne wokół drugiego, aby jak najlepiej zabezpieczyć Twoją cywilizację.

9. Jeżeli w jednym z miast produkujesz oddziały militarne lub cywilne, drugie możesz przeznaczyć na budowę cudów świata.

PAKTY I DANINY

1. Nie bój się odrzucać propozycji innych cywilizacji. Co prawda mogą oni potraktować to jako powód do wojny, jednak mogą też respektować Twoją niezależność i zaferują daninę.

2. Poznaj swoich sąsiadów. Niektórzy mogą być nastawieni pokojowo, jeżeli im zapłacisz, inni zrobią to bezinteresownie.

3. Mówiąc krótko: nie ufaj Stalinowi, Mao, Hammurabiemu i Genghis Khanowi. Stosunek do innych musisz sobie wyrobić na podstawie doświadczeń.

4. Jedna z cywilizacji może zaproponować Ci włączenie Twoich sił militarnych do kampanii przeciwko trzeciej. Musisz bardzo uważać — może się zdarzyć, że masz możliwość rozbicia innego, bardziej rozwiniętego przeciwnika.

5. Pomyśl dwa razy, zanim zapłacisz daninę. Cywilizacje, które żądają od Ciebie pieniędzy, często będą wtrącać się w sprawy Twojego państwa. Zapłać jedynie wtedy, gdy nie masz innego wyjścia.

6. Wymiana zdobyczy naukowo-technologicznych to bardzo ryzykowna rzecz. Wymieniaj się tylko z cywilizacjami bardziej rozwiniętymi, ale słabymi. Rozdawanie wiedzy mocniejszym i wojowniczym nie jest najmądrzejszym posunięciem.

7. Spotykaj się z liderami innych cywilizacji co najmniej przy co trzeciej próbie nawiązania przez nich kontaktu z

Tobą. Zabiera to czas, ale notoryczne unikanie spotkania z innymi przywódcami może być rozumiane jako ignorancja i prowadzi niechybnie do wojny.

8. Nawet posiadanie jednego z cudów świata — Narodów Zjednoczonych (United Nations Wonder) nie może Cię w stu procentach uchronić przed łamaniem paktów o nieagresji, szczególnie w późniejszym stadium gry. Jeżeli nie masz opinii agresora i grasz pokojowo, zaraz po takim podstępym ataku wyślij dyplomatów i rozpocznij negocjacje; wróg zaproponuje zapłacenie przez Ciebie daniny. Jeżeli jesteś nastawiony bardziej wojowniczo, wykorzystaj czas, który dają Ci Narody Zjednoczone, do zgromadzenia armii zdolnej odeprzeć atak i zmiażdżyć przeciwnika.

9. Zwróć uwagę, czy przeciwnik ma broń nuklearną. Niektórzy nie boją się użyć bomby atomowej. Nawet jeżeli przeżyjesz i będziesz w stanie zainicjować negocjacje z agresorem, to zanieczyszczenie środowiska wywołane przez chmury radioaktywne może zrujnować Twój wynik. Najlepszą obroną jest atak — staraj się więc zniszczyć przeciwnika, który ma broń nuklearną, jeżeli oczywiście możesz.

10. Jeżeli masz o wiele od siebie silniejszego wroga, spróbuj stworzyć koalicję przeciwko niemu.

To tyle w tym numerze, dalsza część podpowiedzi w następnym.

Rafał Wiosna

Lp.	Nazwa gry	Nazwa Firmy
1	Civilization	Microprose
2	Pinball Dreams	21st Century
3	Another World	Delphine
4	Kick Off 2	Anco
5	Dune	Virgin
6	Lotus Turbo Challenge II	Gremlin
7	Lemmings	Microprose
8	Birds Of Prey	Electronic Arts
9	Elvira II	Accolade
10	Populous II	Electronic Arts
11	Dungeon Master	FTL
12	Megatraveller	Paragon
13	Secret Of Monkey Island II	Lucasfilm
14	Speedball II	Image Works
15	F19 Stealth Fighter	Microprose
16	Powermonger	Electronic Arts
17	Rock'n Roll	Electronic Arts
18	Sim City	Infogrames
19	North & South	Infogrames
20	Secret Of Monkey Island	Lucasfilm



Dzięki gniazdom, znajdującym się na tylnej ścianie obudowy, Amiga ma kontakt z "resztą świata". Najczęściej wykorzystywanym, ale i najbardziej wrażliwym na uszkodzenia, jest gniazdo równoległe (parallel port). Urządzenie, którego samodzielne wykonanie proponujemy, uczyni Twoją Amigę bezpieczniejszą.

Gdy Amiga zachowuje się niezbyt normalnie (na przykład przestał działać zegar lub drukarka nie reaguje) — oznacza to najprawdopodob-

Przełącznik drukarek i zabezpieczenie gniazda

PODWÓJNE ZABEZPIECZENIE

niej uszkodzenie układu CIA (Complex Interface Adaptor). Układ ten psuje się najczęściej wówczas, gdy przełączasz urządzenia, jakie mają być podpięte do tego gniazda (wszystko jedno, czy będzie to drukar-

ka, digitizer obrazu czy sampler) i zapomnisz przy tym wyłączyć Amigę z sieci. Istnieje wprawdzie możliwość zakupu przełącznika mechanicznego, jednak urządzenie to może również generować szkodliwe

dla układu CIA ładunki elektrostatyczne. Proponujemy Ci przełącznik z zabezpieczeniem elektronicznym. Jest on tańszy od dosyć drogiego (około 350 000 zł) przełącznika mechanicznego i pewniejszy w działaniu.

Proponowany układ pracuje bez zastrzeżeń z A500, A500+, A2000 i A3000. Posiadacze A1000 muszą wykonać łączówkę, gdyż obłożenie złącza gniazda szeregowego jest w ich komputerze nieco inne. Łączówka taka przyda się im

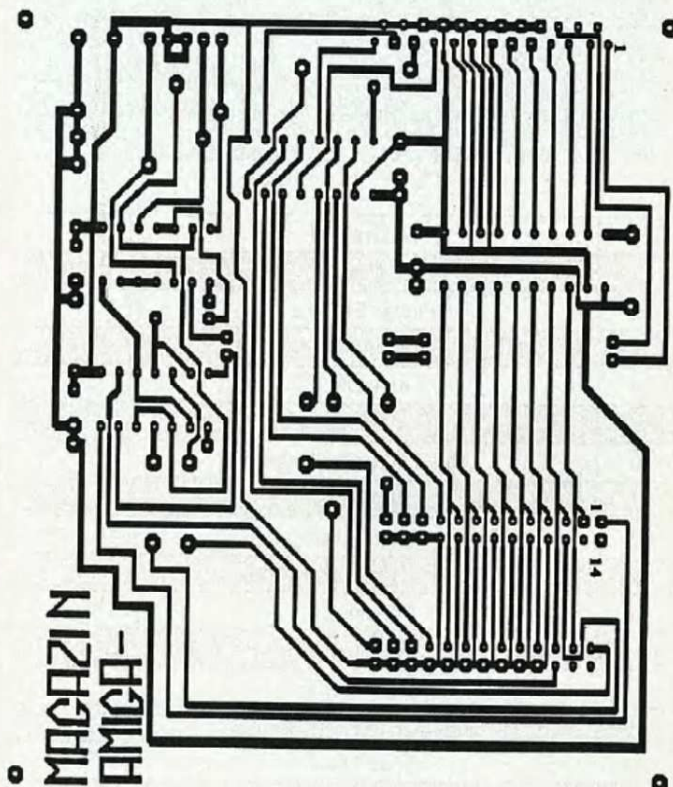
Opis połączeń		
Urządzenie	oznaczenie na płycie	łączyć za pomocą
Amiga	J 1	wtyczka Sub-D 25-złączowa "męska"
Drukarka 1	J 4	gniazdo Sub-D 25-złączowa "żeńska"
Drukarka 2	J 5	gniazdo Sub-D 25-złączowa "żeńska"
Zasilanie	J 3	dwa kołki lutownicze
Przełącznik	I przycisk	łącznik słupkowy 5-złączowy

Spis elementów		
liczba	nazwa elementu	zastosowanie
4	opornik 1 kohm	zabezpieczenie
4	kondensator 100 nF	wyglądzenie napięcia zasilającego
1	układ scalony 74LS00	reset i zabezpieczenie przełącznika
1	układ scalony 74LS92	rozdzielenie sygnałów RESET i STROBE
1	układ scalony 74LS157	sygnał stanu wyboru drukarki
1	układ scalony 74LS245	buforowanie danych
1	przełącznik 1-pozycyjny (włączony/wyłączony)	wyбір drukarki
1	przycisk	reset drukarki
1	wtyczka SubD 25-złączowa męska	podłączenie do gniazda Amigi
2	gniazdo SubD 25-złączowe żeńskie	podłączenie drukarek
1	płytki	wg rysunku obok
1	wtyczka 23-złączowa męska	przy zasilaniu pobieranym z gniazda dodatkowej stacji dysków ALBO
1	zasilacz zewnętrzny	przy zasilaniu zewnętrznym stabilizowany +5V

Jak naświetlić właściwie

Rysunek płytki (obok) został przygotowany w ten sposób, aby można było płytkę bezpośrednio naświetlić. Zauważcie, że na następnej stronie znajduje się w tym miejscu rozmieszczenie elementów w kolorze niebieskim. Nie jest to żaden błąd wydruku, lecz pewien zamysł. Ponieważ kolor niebieski jest dla promieni ultrafioletowych "niewidoczny" — zatem to, co jest narysowane w tym kolorze, nie pojawi się na płycie. Aby prawidłowo wykonać naświetlenie, nie zalecamy kserowania rysunku płytki. Należy wyciąć go nożyczkami. Następnie pokryjcie powierzchnię rysunku (z obu stron) lakierem do włosów w aerozolu. Poczekaście, aby wysychał (zbyt szybkie suszenie spowoduje skurczenie się rysunku, co utrudni montaż). Następnie połóżcie rysunek tak, aby strona zadrukowana na czarno leżała na płycie. Nakryjcie całość cienką płytką szklaną i naświetlajcie lampą kwarcową. Czas naświetlania zależy od rodzaju lampy. Najlepiej wypróbować ten czas na jakimś kawałku płytki.

Od redakcji: Mimo szacunku dla pomysłu kolegów z niemieckiego AMIGA Magazin — zalecamy jednak tradycyjną "fotograficzną" metodę.



Układ Płytki Układ jest przygotowany do naświetlenia stykowego. Połóżcie kartkę na płycie stroną zadrukowaną na czarno w dół. Na gotowej płycie musi być widoczny napis "AMIGA Magazin".



również wówczas, gdy będą chcieli podłączyć do swojej "tysiączki" urządzenie dopasowane do innych Amig. Opis obłożenia złącz w A1000 i w pozostałych Amigach znajduje się w dalszej części tego artykułu.

Wiemy już, że urządzenie ma zabezpieczać gniazdo, lecz po co nam przetłącznik. Przypuśćmy, że mamy drukarkę laserową, która doskonale drukuje grafikę, lecz wykonywanie na niej listingów długich programów mija się z celem (choćby ze względu na koszt kasety z tonerem). O wiele bardziej opłaca się kupić tanią (nawet używaną) drukarkę znakową, która nie musi mieć nawet możliwości graficznych. Do gniazda równoległego możesz także podłączyć digitizer obrazu lub dźwięku (sampler). Stałe przetaczanie tych urządzeń może spowodować mechaniczne uszkodzenie samego gniazda lub (w opisa-

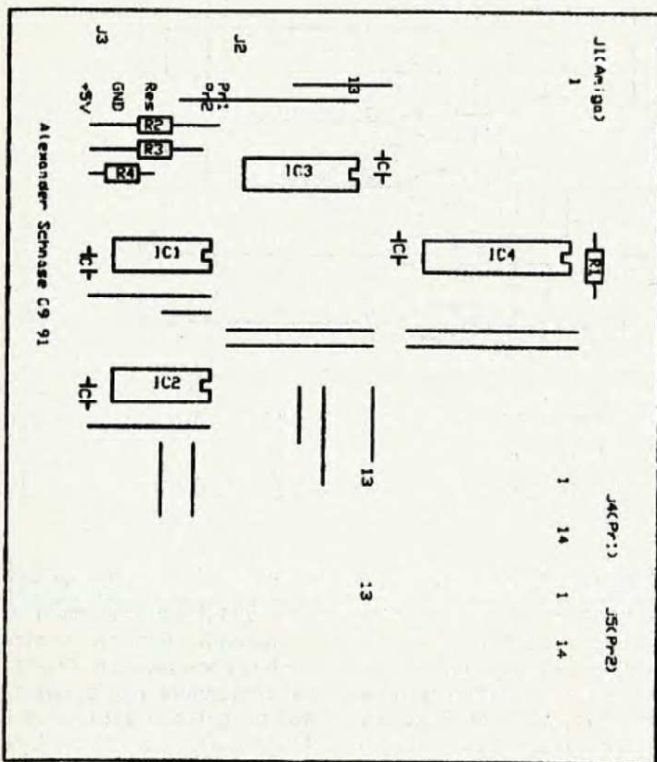
nych na początku warunkach) układu CIA. Od redakcji: wymiana uszkodzonej z winy użytkownika kości CIA zazwyczaj nie jest objęta gwarancją i kosztuje obecnie około 600 000 zł.

UWAGA: Urządzenie skonstruowane według podanego w artykule schematu działa poprawnie tylko w przypadku dwóch drukarek. Oprogramowanie uruchamiające sampler i digitizer zmienia nieco sterowanie układem CIA, zatem opisany układ może w takim przypadku nie zadziałać właściwie. Jeśli Czytelnicy wykazą duże zainteresowanie "złodziejką" drukarka/sampler, wówczas postaramy się opublikować schemat i opis wykonania takiego urządzenia. Po jego wykonaniu będziesz miał opisane wyżej kłopoty "z głowy". Zasada działania jest następująca: Dane z Amigi są podawane obu drukarkom

przez układ sterowania danych (74LS245). Za pomocą przetłącznika można wybrać drukarkę, której chcesz używać. Sygnał zmiany zostanie przepuszczony przez dwie połączone krzyżowo bramki NAND (74LS00). "Zabezpieczacz" przepuści sygnał STROBE wskazujący na gotowość do przesyłania danych tylko wybranej drukarce. Meldunki zwrotne (BUSY, ACK, POUT i SELECT) zostaną przetworzone za pomocą multiplexera (74LS157). Przycisk pozwala na zresetowanie aktualnie wybranej drukarki bez konieczności resetowania komputera. Jest to konieczne na przykład wtedy, gdy w buforze drukarki znajdują się jeszcze wprowadzone tam poprzednio dane, których nie chcesz drukować. Bity danych (D0 do D7) są buforowane przez układ 74LS245 i przesyłane stamtąd na wejście wybranej

nie ma, zatem osobom nie mającym zbyt dużego doświadczenia w elektronice nie polecamy tego rozwiązania. Lepiej jest pobrać napięcie z gniazda dodatkowej stacji dysków. Nawet jeśli jest tam podłączona jakaś stacja, to większość z nich ma gniazdo przelotowe. Zasilanie +5V pobieramy ze złącza 12. gniazda stacji dysków (lub "przelotki"). Rozwiązanie to ma jednak pewną wadę. Należy dodatkowo zakupić wtyczkę 23-złączową (a w Polsce są z tym problemy — przyp. tłum.), a ponadto przy równoczesnej pracy przetłącznika i dodatkowej stacji dysków mogą wystąpić zakłócenia pracy tej ostatniej.

Najlepszym rozwiązaniem, zwłaszcza dla użytkowników A500 i A500+, w których zasilacz jest zbyt "wyżyłowany" i podłączenie każdego dodatkowego urządzenia wy-



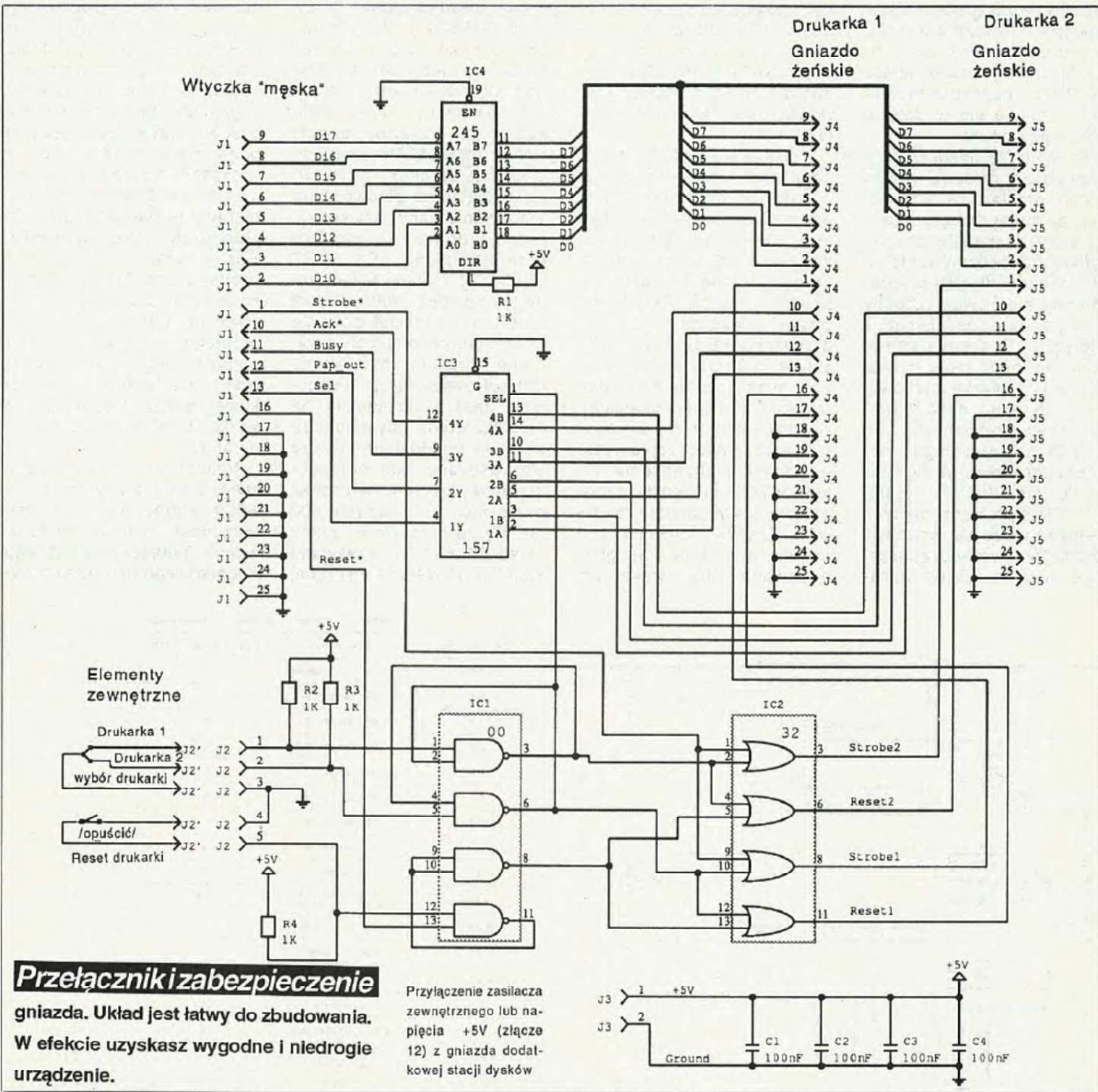
Rozmieszczenie elementów Zacznijcie wlotowywać na płytkę elementy najtańsze i najbardziej odporne. Pod układy zastosujcie bezwarunkowo podstawki. Zwróćcie uwagę na prawidłowe włożenie układów do podstawek.

Obłożenie styków gniazda równoległego			
Oznaczenie	Znaczenie	A500/500+/2000/300	A1000
STROBE	Gotowość danych	1	1
D 0	bit danych 0	2	2
D 1	bit danych 1	3	3
D 2	bit danych 2	4	4
D 3	bit danych 3	5	5
D 4	bit danych 4	6	6
D 5	bit danych 5	7	7
D 6	bit danych 6	8	8
D 7	bit danych 7	9	9
ACK	potwierdzenie odbioru danych	10	10
BUSY	drukarka zajęta	11	11
POUT	brak papieru	12	12
SELECT	drukarka jest ON LINE	13	13
+ 5V	zasilanie	14	23
.	styk nieobłożony	15	24
RESET	potwierdzenie zwrotne	16	25
GND	masa	17-25	14-22

drukarki. Bramka OR (74LS32) pozwala na skierowanie sygnałów STROBE i RESET wyłącznie na wybraną drukarkę. Impuls otrzymany z bramek NAND jest doprowadzony do bramek OR, gdzie STROBE i RESET zostają rozdzielone. Zasilanie urządzenia może być pobrane bezpośrednio z Amigi. Podłączenie napięcia +5V z gniazda parallel port możliwe jest jedynie wówczas, gdy na złączu tym wmontowany jest opornik ograniczający. W modelu fabrycznym takiego opornika tam

magającego zasilania może spowodować problemy — jest zasilanie zewnętrzne. Przy budowie, lub zakupie, dodatkowego zasilacza należy zwrócić uwagę na to, aby był to bezwzględnie zasilacz stabilizowany. Przy nie stabilizowanym napięciu mogą bowiem wystąpić prądy błądzące, co nie wyjdzie na zdrowie naszemu urządzeniu.

Płytkę na Amidze została zaprojektowana programem o nazwie "L100". Jeśli macie taki program (dostępny jest on na dyskietce z programami



Przełącznik i zabezpieczenie gniazda. Układ jest łatwy do zbudowania. W efekcie uzyskasz wygodne i niedrogie urządzenie.

Przyłączenie zasilacza zewnętrznego lub napięcia +5V (złącze 12) z gniazda dodatkowej stacji dyskowej

z numeru 12/91 niemieckiego AMIGA Magazin — przyp. tłum.), możecie dowolnie zmienić płytkę, aby dopasować ją do swoich potrzeb. Aby nie było problemów z budową urządzenia, zaprojektowano druk jednostronny. W ten sposób utracono możliwość wykonania niektórych potrzebnych połączeń za pomocą druku po drugiej stronie. W związku z tym należy włutować po drugiej stronie "mostki" wykonane z drutu (na

rysunku zestawienia elementów są to kreski). Lutowanie rozpoczynamy od najmniej wrażliwych elementów, a zatem włutowujemy w kolejności: wspomniane wyżej mostki, przełącznik, przycisk, gniazdko wyjściowe i wejściowe, oporniki, kondensatory i podstawki pod układy. Przy wkładaniu układów scalonych do podstawek zwróćcie uwagę na ukierunkowanie nacięć na układach. Powinny być one takie, jak na rysun-

ku zestawienia części. Przy próbach można gniazda wyjść drukarki przyłączyć za pomocą kabli wstążkowych zakończonych 26-złączowymi wtyczkami typu Scotchflex (podraża to jednak koszt budowy o około 150.000 zł — przyp. tłum.). Po włutowaniu wszystkich części obejrzyjcie dokładnie płytkę, sprawdźcie, czy nie ma na niej zwarów spowodowanych zbyt "obfitym w cynę" lutowaniem. Skontrolujcie także prawidłowość

wbudowania elementów i ułożenia układów scalonych w podstawkach. Wtyczkę urządzenia należy wkładać do gniazda parallel port Amigi wyłącznie wtedy, gdy komputer jest wyłączony z sieci.

Amiga Magazin 12/1991
Tłum. Marek Pampuch
Od redakcji: Wydaję nam się, że nie ma w tej chwili w Polsce firmy produkującej "złocijki do gniazda równoległego". Jeżeli się mylimy, prosimy zainteresowanych o nadślanie sprostowania na adres redakcji.



W każdym użytkowniku Amigi drzemią ukryte możliwości. Każdy na pewno potrafi zrobić ze swoim komputerem coś, czego nie umieją inni, choć być może nawet nie zdaje sobie z tego sprawy. A spróbować warto!

Jest już odzew na nasz apel o nadsyłanie sztuczek. Tricky dostarczone przez Czytelników będą systematycznie zamieszczane w "Kufereku", a zgodnie z naszą obietnicą, ich autorzy otrzymają cztery dyskietki. Aby odkryć sztuczkę, nie potrzeba wcale wielkich umiejętności w zakresie obsługi komputera. Najczęściej odkrycie jest kwestią przypadku, tak jak w przypadku jednej z nadesłanych sztuczek, której autor wpadł na pomysł po zaledwie trzech dniach używania Amigi. Zgodnie z zasadą, na jakiej funkcjonuje "Kuferek", nie musi to być sztuczka programowa czy sprzętowa. Wystarczy opis rozwiązania problemu, z którym daliście sobie radę. Przy okazji prosba. Ponieważ otrzymujemy coraz więcej listów na różne tematy — na listach i kartkach dotyczących tej rubryki zamieszczajcie dopisek "AMIGA-Kuferek".

* *Obraz monochromatyczny*

Tomasz Stasiak

Na odpowiednio podłączonym monitorze kolorowym (na przykład 1084 S) można wyświetlić obraz monochromatyczny (odcienie szarości). Jak wiadomo, Amiga ma wyjście monochromatycznego sygnału wizyjnego (jeśli patrzymy z tyłu na A500, jest to pierwsze gniazdo z prawej strony). Monitor, którego używamy, musi mieć wejście takiego sygnału. Mają je firmowe monitory C 1084 lub C 1084 S, ewentualnie inne

monitory z wejściem composite video (w 1084 S wejście to nazywa się CVBS). Kablem koncentrycznym (ale nie antenowym) z wtyczkami typu chinch łączymy oba gniazda. Jeśli nie mamy takiego kabla, możemy sami go wykonać albo poświęcić kabel od naszego zestawu wieżowego. Pozostaje jeszcze tylko przełączenie monitora tak, aby odbierał sygnał mono. W przypadku monitora 1084 S należy w tym celu przycisnąć przycisk CVBS/RGB. Jeśli chcemy znów pracować w trybie kolorowym, przyciskamy jeszcze raz ten sam przycisk i już mamy kolor.

Monochromatyczny obraz uzyskiwany po takiej operacji jest wyraźniejszy od obrazu kolorowego i mniej męczy oczy. Doskonale nadaje się do prostych czarno-białych prac DTP czy do pisania tekstów.

* *SEKA i drukarka*

Frank Schneider

Jeśli chcemy wydrukować kod źródłowy uzyskany za pomocą assemblera SEKA, wówczas za pomocą:

```
> [Enter] nazwa> prt:
```

(gdzie parametr nazwa jest nazwą zbioru z naszym kodem źródłowym) możemy uzyskać jego wydruk. Nie zawsze jednak potrzebny jest cały wydruk. Wykorzystajmy zatem rozkaz programu:

```
p nn
```

(gdzie nn jest liczbą wierszy, jakie chcemy wydrukować, liczoną od wiersza aktualnego). Po zakończeniu wydruku należy nacisnąć "pusty" klawisz [Enter], co opróżni bufor drukarki i zamknie kanał wydruku.

Amiga Magazin 8/1991

Tłum. Marek Pampuch

** *Najprostsze śledzenie programu w assemblerze*

Frank Schneider

Podczas testowania dużych programów napisanych w assemblerze niekiedy trzeba sprawdzić tylko jakąś część tego programu. Pracę tę można sobie ułatwić — wpisując, w wierszu poprzedzającym początek testowanego fragmentu, rozkaz:

```
move #50kol,$dff180
```

co zmieni kolor przeszukiwanego fragmentu programu na

ekranie, pod warunkiem, że zamiast literek "kol" wpisujemy tutaj kod heksadecymalny z zakresu od 0 do F — odpowiada jący jednemu z 16 kolorów.

Amiga Magazin 8/1991
Tłum. Marek Pampuch

*** Surowe teksty

Ilse Koch

Pożytecznym narzędziem przy programowaniu w assemblerze jest funkcja "RawDoFmt". Programiści korzystają z niej chętnie, ponieważ działa prawie tak samo jak funkcja "printf()" z języka C. Funkcji tej należy określić parametry dotyczące łańcucha formatowania i pola danych. Trzeba także napisać niewielką procedurę, która będzie odczytywać kolejno pojedyncze znaki wydruku tekstu. Łańcuch formatowania może, oprócz normalnych znaków, zawierać także znak rozkazu formatowania "%" znany tym wszystkim, którzy programują w języku C. W przypadku, gdy chcemy użyć symbolu "%" jako znaku w łańcuchu, wówczas rozkaz "%" należy zastąpić rozkazem "%%". Rozkaz "%" ma następujące opcje:

```
%[-] [szerokość] [limit] [l] [d|x|s|c]
```

Za pomocą rozkazu formatowania można napisać zarówno liczby dziesiętne, jak i heksadecymalne, a także pojedyncze znaki. Liczby dziesiętne i heksadecymalne można wyprowadzić w postaci słów lub w postaci długich słów. Wszystkie wydruki da się justować zarówno do prawej, jak i do lewej strony. Drukowane liczby można dopełnić zerami na początku liczby. Przy łańcuchach znakowych można przy tym ustalić wielkość pola wydruku. Przejdźmy teraz do omówienia opcji:

[%] = rozkaz formatowania

[-] - oznacza, że wydruk będzie justowany po lewej stronie [szerokość] jest to liczba dziesiętna, która określa szerokość pola wydruku. Jeśli parametr ["szerokość"] nie jest poprzedzony znakiem "minus", wówczas wydruk będzie justowany tylko po prawej stronie. Jeżeli łańcuch znaków jest dłuższy, aniżeli podana wyżej szerokość, to nie ulegnie ona skróceniu, lecz wydruk wyjdzie poza oznaczone pole. Aby tego uniknąć, należy użyć parametru "limit", o którym będzie mowa dalej. Jeśli wprowadzamy liczby z zerem, wówczas zera należy wprowadzić zarówno przy zapisie dziesiętnym, jak i heksadecymalnym. [Limit] — jest to liczba dziesiętna, która określa dopuszczalną szerokość wydruku. Opcja ta jest pożyteczna zwłaszcza wtedy, gdy użyjemy jej razem z parametrem "szerokość". Uwaga: "szerokość" i "limit" nie mogą być wprowadzane w postaci binarnej, lecz muszą być podane jako

tekst. Jeżeli zastosujemy parametr "l", wówczas liczby dziesiętne i heksadecymalne muszą być podawane w formacie długich słów.

Kolejnym parametrem jest jeden ze znaków d, x, s lub c. Parametr ten określa sposób wydruku:

- d = liczba dziesiętna;
- x = liczba heksadecymalna;
- s = łańcuch znakowy;
- c = znak.

Przykładowo może to być:

- %x = liczba heksadecymalna w formacie słowa;
- %ld = liczba dziesiętna w formacie długiego słowa;
- %12c = znak będzie wydrukowany w polu o długości 12 znaków i justowany po prawej stronie;
- %12s = łańcuch znaków będzie wydrukowany w polu o długości 12 znaków, i justowany poprawiej stronie;
- %-12.10s = justowanie następuje po lewej stronie w polu obejmującym 12 znaków. Jednak gdy łańcuch będzie dłuższy niż 10 znaków, wszystkie nadmiarowe znaki przy wydruku zostaną obcięte;
- %08lx = liczba heksadecymalna wydrukowana zostanie w postaci 8 znaków i ewentualnie poprzedzona zerami.

Oto przykład łańcucha formatowania:

```
Lancuch: dc.b "Szanowny Panie %s. Informujemy ze zalega Pan od dnia"
dc.b "%02dd.%02d.%4d na kwote %ld złotych.",0
```

oraz związanego z nim rekordu danych:

```
Dane: dc.l Nazwisko
dc.w 13.06.1992
dc.l 2545700
Nazwisko:
dc.b "Przykładowski"
```

W tym przypadku dzięki funkcji "RawDoFmt" otrzymamy następujący wydruk:

```
Szanowny Panie Przykładowski. Informujemy ze zalega Pan od dnia
13.06.1992 na kwote 2545700 złotych.
```

Pole danych zawiera dane dla wydruku. Muszą one występować w polu danych w tej kolejności, w jakiej będą wykorzystywane w łańcuchu formatowania. Liczby dziesiętne i heksadecymalne zamieszczone w polu danych muszą, zależnie od formatu, występować w formie słów albo długich słów. W przypadku łańcucha znaków, pętla musi mieć etykietę. Łańcuch alfanumeryczny, podobnie jak i łańcuch formatowania, musi być zakończony zerem.

A teraz zilustrujemy całość zagadnienia za pomocą przykładowego programu.

Uwaga: Funkcja RawDoFmt biblioteki Exec wymaga co najmniej trzech parametrów, jednak wprowadzenie ewentualnego czwartego parametru nie ma żadnego wpływu na wykonanie funkcji; parametr ten jest tylko przekazywany dalej do funkcji przejmującej sformatowany już łańcuch.



** Trafic (z tekstem) w dziesiątkę

Jens Freisen

Każdy, kto usiłował programować grafikę w Amiga BASIC-u — zapewne często złościł się z powodu braku możliwości dokładnego umieszczenia tekstu na rysunku. Istnieje wprawdzie rozkaz LOCATE, ale pozwala on jedynie na pozycjonowanie tekstu z dokładnością do jednego wiersza lub kolumny. Na szczęście Amiga daje nam o wiele większe możliwości niż to, co proponują "geniusze" ze stajni Billa Gatesa. Wykorzystajmy zatem umiejętności naszego komputera. W naszym przypadku posłużymy się rozkazami z biblioteki "graphics.library". Znajduje się w niej rozkaz "Move&Rastport&,x&,y&". Jeśli określimy strukturę portu rastera okna przez funkcję BASIC-a WINDOW z parametrem 8, wówczas rozkaz "Move&Rastport" umieści nam kursor, zarówno tekstowy, jak i graficzny, w punkcie o współrzędnych x& i y&. Aby móc skorzystać z dobrodziejstw takiego rozwiązania, należy się najpierw upewnić, czy mamy na dysku zbiór danych z odpowiednią mapą bitów ("bmap"), a ponadto, na początku programu należy otworzyć rozkazem LIBRARY bibliotekę. Rozkaz biblioteczny można wywołać bezpośrednio (odpowiednim rozkazem BASIC-a). My jednak posłużymy się tu funkcją POKE. W związku z powyższym — podprogram, ustawiający kursor, będzie wyglądał jak poniżej:

```
SUB TextXY (X&,Y&) STATIC
POKEW WINDOW(8)+36,X&
POKEW WINDOW(8)+38,Y&
ENDSUB
```

Teraz wystarczy jeszcze tylko obliczyć lub wprowadzić współrzędne X& i Y&, a następnie wywołać podprogram rozkazem:

```
TextXY (X&,Y&).
```

Prostszego sposobu prawdopodobnie nie ma. (A może któryś z Czytelników jest odmiennego zdania? Czekamy na odzew — red.)

AMIGA Magazin 8/1991
Tłum. Marek Pampuch

```
SysBase EQU 4
start:
lea Napis(pc),a0
bsr.w printf
moveq #0,d0
rts
Napis:
dc.b "Komputer %s produkowany przez ",34
dc.b "%c",34." ma standardowo montowany procesor",10
dc.b "o symbolu %ld produkcji koncernu
dc.b "Pamięć ta w tym modelu rozciąga się od "
dc.b "adresu %06lx do %06lx.",10,0
```

```
even
dc.l Komputer
dc.w "C"
dc.l 68000
dc.l Firma
dc.w 512
dc.l 0,$7ffff
Komputer:
dc.b "Amiga",0
even
Firma:
dc.b "Motorola",0
even
printf:
movem.l d0-d7/a1-a6,-(sp) ; a0 = adres tekstu
move.l a0,a1
loop:
tst.b (a1)+
bne.s loop
move.l a1,d0
addq.l #1,d0
bclr #0,d0
move.l d0,a1 ; a1 = tekst do uzupełnienia
lea putchproc(pc),a2 ; a2 = adres procedury wypisujacej
lea putchdata(pc),a3 ; a3 = adres bufora tekstu
clr.l lenght
move.l (SysBase).w,a6
jsr -522(a6) ; RawDoFmt
lea DosName(pc),a1 ; a1 = nazwa dos.library
moveq #0,d0
jsr -552(a6) ; OpenLibrary
tst.l d0
beq.s error1
move.l d0,a6
jsr -60(a6) ; Output
move.l d0,d1
beq.s error2
move.l #putchdata,d2
move.l lenght(pc),d3
jsr -48(a6) ; Write
error2:
move.l a6,a1
move.l (SysBase).w,a6
jsr -414(a6) ; CloseLibrary
error1:
movem.l (sp)+,d0-d7/a1-a6
rts
putchproc:
move.b d0,(a3)+
addq.l #1,lenght
rts
putchdata: ds.b 256 ; dlugosc wg uznania
lenght: dc.l 0
DosName: dc.b "dos.library",0
even
```

RawDoFmt.asm stworzy sformatowany wydruk.

TANIE I LEGALNE

Dzisiaj kolejna porcja "rybnych smakołyków". Jak zapewne zauważycie, niektórzy autorzy nadsyłają do Freda nowe wersje przedstawianych wcześniej programów.

Axel Winzer

FISH DISK 512

CSH — jest programem zastępującym Shell Amigi zbliżonym w możliwościach do Unixowskiego CSH. Mamy tutaj ponad 100 zintegrowanych rozkazów, 70 różnych funkcji, dodatkowe zmienne systemowe, kompleksowe nazwy zbiorów danych, automatyczną zmianę katalogów, menu sterowane za pomocą myszki i wiele innych. Wyjątkowo duża szybkość, z jaką pracuje ten program, nie pozostawia wyboru: należy go koniecznie mieć. Wersja 5.15. Rozszerzenie wersji 4.02 z Fish-dysku 458. [Program i kod źródłowy]. Autorzy: Dominik Mueller, C.Borrore, S.Drew i M. Dillon.

FLIPIT — Kolejna wersja gry Reversi (otello), w której komputer jest bardzo trudnym przeciwnikiem. [Program.] Autor: Adrian Millett.

M2PASCAL — Nazwa wyraźnie wskazuje na to, że program ten służy do tłumaczenia na Pascal innych programów napisanych w języku Modula-2. W ten sposób można na przykład program, stworzony w jednym z licznych kompilatorów Moduli na Amidze, przenieść na kod źródłowy w Pascalu i pracować na nim dalej pod kompilatorem Pascala (nawet na innych komputerach!). Wersja 1.0. [Program i kod źródłowy w Moduli-2.] Autor: Greg Mumm.

SOLITAIRE — Kolejna odmiana gry "Samotnik" (nieco inna niż ta z dysku 511). Zaletami jej są ładna grafika i opracowanie muzyczne [Program.] Autor: Pat Clark.

FISH DISK 513

DKBTRACE (DKB) — jest chyba najlepszym programem do raytracingu, jaki można uzyskać w kategorii programów niekomercyjnych. W porównaniu z programami, za które płacisz ciężkie pieniądze — nie jest wcale gorszy, i to z jakiegoś kolwiek punktu widzenia. Program wczytuje zwykłe dane zapisane w kodach ASCII i przelicza je za pomocą posiadanych rozkazów na kompletny rysunek. Potrafi nawet pracować z obrazami grafiki 24-bitowej (jeśli nie masz karty grafiki 24-bitowej, wówczas obrazy te muszą być wcześniej "przełożone" na tryb HAM). Różne struktury i duża liczba form pozwalają na wielostronne opracowywanie obrazów. Dodatkowo masz dostęp do licznych danych przykładowych, które mogą być potraktowane jako podstawa do pierwszych "prywatnych" obliczeń. Program jest dosyć duży, zatem jego "reszta" musiała być przerzucona na dysk 514. Wersja 2.12. Rozszerzenie wersji 2.0 z Fish-dysku 397 [Program.] Autor: David Buck.

NEWLIST — Procedura zastępująca rozkaz LIST, która zawiera wiele dodatkowych funkcji. Mamy tu na przykład: filtr, rozróżnianie dużych i małych liter w nazwach zbiorów, filtry Uniksowe, większość funkcji oryginalnego rozkazu LIST i wiele innych. Z nowych funkcji godne

podkreślenia są: możliwość obróbki wielu ścieżek oraz zatrzymywanie wydruku po każdej stronie ekranowej. Wersja 5.0a. Rozszerzenie wersji 5.0 z Fish-dysku 501. [Program] Autor: Phil Dietz.

FISH DISK 514

DKBTRACE — Dokończenie programu DKB trace z dyskietki 513. Właściwie nie tyle dokończenie, ile rozmaite programy pomocnicze. Można za ich pomocą przykładowo stworzyć nowe zbiory danych lub przystosować już istniejące zbiory do przeliczenia. [Program] Autor: David Buck.

GEARCALC — Jest to program pozwalający na przeliczenie przełożenia transmisji kół samochodu (??? — przyp. red). Wersja 2.0 [Program.] Autor: Ed Bacon.

S220T08SVX — Program ten przekłada sample z Rolanda S220 lub S10 czy MKS-100 na dobrze wszystkim muzykom amigowskim znany format IFF-8SVX (8-bitowy). Ta wersja pozwala na nastawienie wzmocnienia, a także na zaznaczenie punktu początkowego i końcowego przetwarzania. Wersja 1.4. Rozszerzenie wersji 1.0 z dysku Fisha n286. [Program i kod źródłowy.] Autor: Dieter Bruns.

TLOG — Program sterowany za pomocą myszki, który tworzy zbiory statystyczne opisujące nasze treningi. Można tu wpisać wartości do dotyczące trasy, czasu, rytmu serca, wagi i temperatury. Za pomocą rozkazów Arexxa można łatwo ingerować w program. Wersja 1.0 (Shareware). [Program.] Autor: Ed Bacon.

FISH DISK 515

CHECKBOOK — Procedura służąca do wyświetlania, poprawiania i analizy aktualnego stanu naszego konta. Wprawdzie nie zastępuje książeczki czekowej, lecz działanie programu jest na niej oparte i może być pomocne w dokonywaniu wielu transakcji. Można sortować, kopiować, poprawiać dane, a także przeprowadzać badania statystyczne na wprowadzonych danych. Program będzie działał prawidłowo z systemem operacyjnym w wersji 1.3 lub 2.0. [Program.] Autor: Jeffrey R. Almasol.

D110EDDEMO — Jest to wersja demonstracyjna programu pozwalającego na edycję sampli uzyskanych za pomocą syntezatora Roland D-110. [Program demonstracyjny.] Autor: Dieter Bruns.

PP — Power Patcher jest programem, który chętnie miałoby wielu użytkowników Amigi. Skompresowane dobrze znanym Power Packerem dane są niedostępne, bez uprzedniego ich zdekranczowania. Power Patcher pozwala na używanie takich danych w postaci skompresowanej, wykorzystując w tym celu bibliotekę DOS.Library. Pozwala on także na archiwizację zbiorów opisujących ikony (z rozszerzeniem ".info"). Umożliwia również obróbkę skompresowanych tekstów za pomocą dowolnego edytora. Wersja 1.0 (shareware). [Program i kod źródłowy.] Autor: Michael Berg.

SETCCOPTS — Umożliwia łatwą pracę z manxowskimi zmiennymi CCOPTS. Można zapamiętać różne ustawienia. Sterowany myszką. Wersja 2.0. [Program.] Autor: Stephan Flother.

FISH DISK 516

ENIGMA — Gra Enigma daje Wam "interaktywną animację", która może być uważana także za układankę typu puzzle. "Maszyna szyfrująca" Enigma może być też przez użytkownika ustawiona w taki sposób, aby tworzyła zaszyfrowane teksty. Wprowadzonych tekstów program użyje tu w celu stworzenia z nich innych "rozsądnie" wyglądających słów. Wersja 1.0. [Program.] Autor: Martin C. Kees.

LOOM — Symulacja warsztatu tkackiego, który używa 15 kolorów nici. Można uzyskać olbrzymią liczbę wielobarwnych "tkanin", zaś wynik pracy można zapisać na dyskietce lub wydrukować. Wersja 1.0. [Program.] Autor: Martin C. Kees.

PHONEGRAM — Procedura tworzy tekst z... numerów telefonów. Program próbuje znaleźć wszystkie słowa trzy- lub czteroliterowe, jakie można zakodować pod numerem telefonu. Wersja 1.01 (Od redakcji: W wielu krajach zachodnich na tarczy telefonu znajdują się nie tylko cyfry od 0-9, ale i litery alfabetu.) [Program.] Autor: Martin C. Kees.



QUOTES — Program, który losowo tworzy cytaty. Wystarczy nim potraktować na przykład dowolny fragment "Trylogii" zapisany jako zbiór ASCII, a Quotes zrobi z niego taką "jajecznicę", że biedny Sienkiewicz chyba przewróci się w grobie. Program potrafi wystartować jedynie z poziomu CLI, a zatem radzimy doinstalować mu odpowiednią sekwencję startową. [Program i kod źródłowy.] Autor: Adam Evans.

REXXVIEW — REXXView dozoruje komunikaty, które zostały pobrane z portu ARexxa. Wszystkie meldunki są podawane razem z numerem aktualnego zadania, podane są także inne informacje o nich. [Program i kod źródłowy w JForth.] Autor: Martin C. Kees.

XI — Wersja demonstracyjna programu, który zdaniem autora jest "generatorem animacji komórkowej opartym na pojedynczym bit-planie". Liczne rozkazy Arexxa i wiele funkcji ułatwiają obsługę programu. Program wymaga biblioteki "req.library". Wersja demonstracyjna nie ma możliwości zapisywania stworzonych dzięki niej animacji. [Program demonstracyjny.] Autor: Martin C. Kees.

FISH DISK 517

AQUEIPOT — program ten służy do przeliczania wielokolorowych rysunków za pomocą specjalnego algorytmu. Pracuje zarówno z rysunkami tworzonymi w trybie PAL, jak i w NRSC, zarówno w wysokiej, jak i w niskiej rozdzielczości. Program ma dwie prędkości działania i (związane z tym) dwie jakości otrzymanego rysunku. Gotowe obrazy można zapisać w formacie IFF. Specjalne skrypty pozwalają nawet na przeliczenie pełnej animacji. Dokumentacja programu jest zarówno po angielsku, jak i po niemiecku. Wersja 1.15. Rozszerzenie wersji 1.06 z dysku 474. [Program i kod źródłowy [PCQ.]] Autor: Juergen Matern.

AMIBACK — jest wersją demonstracyjną nowoczesnego programu do archiwizacji twardego dysku. Wśród jego funkcji znajduje się m.in. zapis na wszystkich możliwych urządzeniach (dyskietki, twarde dyski, dyski wymienne [Bernoulli], Streamer). Użytkownik może wybrać pomiędzy całościową kopią bezpieczeństwa i archiwizacją oznaczonych zbiorów danych. Wersja demonstracyjna nie pozwala jednak na używanie tak zarchiwizowanych zbiorów, ani też na sprawdzenie poprawności archiwizacji. Wersja 1.04. Rozszerzenie wersji 1.03 z dysku 493 [Program demonstracyjny]. Autor: MoonLighter Software.

CWTOY — program z rodzaju "mówiących zabawek", który pomaga komunikować się Amidze w międzynarodowym kodzie Morse'a (tylko z kim? — przyp. tłum.). Wiele użytecznych funkcji sprawia, że program ten będzie właściwym wyborem wszystkich interesujących się językiem Morse'a. Wersja 1.0 [Program i kod źródłowy w C.] Autor: Rob Frohne.

HEX — jak nazwa wystarczająco objaśnia, ten edytor "szesnastkowy" pozwala na tworzenie i obróbkę danych binarnych. Duża liczba funkcji umożliwia wygodną pracę z tym programem. Wersja 1.0 [Program Shareware]. Autor: Nicola Salmoria.

RUSSIANFONTS — ten katalog zawiera czcionki rosyjskie (cyrylicę) w wielkościach od 13 do 31 punktów ekranowych. [Katalog: (fonts).] Autor: Daniel Amor.

SPLINER — wygaszanie ekranu dla systemu operacyjnego w wersji 2.0. [Program.] Autorzy: Tom Rokicki i Sebastiano Vigna.

FISH DISK 518

AMIDOCK — implementacja funkcji "Dock" z komputera NeXT. Program otwiera pod Workbenchem małe okno, które jest wypełnione pędzlem (brush) IFF. Użytkownik może teraz przez wybranie myszką tej ikony (dwuklik) uruchomić dowolny program użytkowy. Wersja 1.3. Rozszerzenie wersji 1.2.4 z dysku 474 [Program Shareware.] Autor: Gary Knight.

LISTER — program do pokazywania różnych informacji zapisanych w zarchiwizowanych zbiorach danych. Potrafi rozpoznać następujące formaty kompresowania danych: "Arc", "Cpio", "LHarc", "Tar", "PkZip" i "Zoo". Wersja 1.0 [Program i kod źródłowy.] Autorzy: Kerry Clanos i Geoffrey Faivre-Mailloy.

POST — komfortowy interpreter do stworzonego przez firmę Adobe języka programowania "PostScript". Program współpracuje z członkami Post Scriptowymi typu 1 i 3, wyniki wyprowadzane są na ekran, do

zbioru danych i oczywiście na drukarkę. Wymaga biblioteki "arp.library" (w wersji 39 lub wyższej) oraz programu "Conman" (w wersji 1.3 lub wyższej). Wersja 1.6. Rozszerzenie wersji 1.5 z dysku 468. [Program i kod źródłowy w C.] Autor: Adrian Aylward.

FISH DISK 519

AVLSORT — programem AVLSort można sortować pojedyncze dane tekstowe. Bazuje on na załączonym na dyskietce pakiecie AVL. Liczba wierszy tekstu, jakie mogą być opracowane tym programem, zależy jedynie od ilości wolnej pamięci. [Program.] Autorzy: Robert Pyron i Mark Mallett.

CHKFRAG — program tworzy raport o fragmentacji pamięci w systemie [Program]. Autor: Timeus.

FIFOLIB — FIFO (first in — first out) bazuje na bibliotece "fifo.library" i pozwala na elastyczną pracę ze stosem. Wersja 3.1. Rozszerzenie wersji 2 z dysku 448 [Biblioteka i jej (niepełny) kod źródłowy.] Autor: Matt Dillon.

OAKLISP — implementacja systemu OAKLISP na Amigę. OAKLISP jest w zasadzie uproszczoną odmianą języka programowania LISP, z tym, że jest nieco bardziej zorientowany obiektowo. Z powodu objętości OAKLISP musiał zostać podzielony na dwa dyski (Fish Disk 519 i 520). Do pracy potrzebne są oba dyski. Autorami systemu są Kevin Lang i Barak Pearlmutter [Program i kod źródłowy]. Implementacja na Amigę: Mike Meyer.

OPTIMIZER — program do optymalizacji zbiorów na dyskietkach i twardej dyskach. Jego zadaniem jest wczytanie rozrzuconych na tych dyskach danych i ponowne zapisanie tych danych, lecz "we właściwym z punktu widzenia Amiga DOS porządku", co w przyszłości znacznie skróci czas ładowania programów (i oszczędzi głowicę dysku — przyp. tłum.). Przy przepisywaniu "bezpieczeństwo" danym zapewnia to, że wciąż zapisywany jest tylko jeden blok. Wersja 10 [Program.] Autor: Tom Stotelmeyer.

FISH DISK 520

IOBOARD — kompletny schemat urządzenia pozwalającego na dołączenie dodatkowych gniazd interfejsów: szeregowego i równoległego do Amigi 500, 1000 i 2000. W przyszłości zapowiadane jest urządzenie pozwalające zwiększyć liczbę gniazd do 4 każdego typu. Zawiera sterowniki pozwalające na obsługę obu tych gniazd. Wersja 2.10. [Schemat, sterowniki oraz ich kod źródłowy.] Autorzy: Jeff Lavin, Dan Babcock i Paul Coward.

OAKLISP — druga część systemu OAKLISP w wersji na Amigę, którego część pierwsza, niezbędnie potrzebna do pracy, znajduje się na dysku 519. [Program i kod źródłowy.] Autorami są Kevin Lang i Barak Pearlmutter. Implementacja na Amigę: Mike Meyer.

FISH DISK 521

A68K — assembler na procesorze Motorola 68000 w wersji "pierwotnej" napisany przez Briana Andersona w języku "Modula-2". W 1987 Charlie Gibb przełożył go na język C. Program w tej wersji akceptuje kod assemblera zgodny ze standardem firmy Metacomco i generuje zbiory "object" na Amigę. Wersja 2.71. Rozszerzenie wersji 2.61 z dysku 314. [Program i kod źródłowy w języku C.] Autor: Brian Anderson, konwersja na język C: Charlie Gibb.

BATCHMASTER — ten program pracuje tak, jakby był kombinacją rozkazów Amiga DOS: Ask, If i Skip. Za jego pomocą można umieścić w zbiorach wsadowych (np. typu "startup-sequence") pytanie z czterema możliwościami odpowiedzi do wyboru. Program wymaga biblioteki "arp.library". Wersja 1.27. [Rozkaz ADOS.] Autor: Janne Pelkonen.

CHECKPRNT — za pomocą własnych rozkazów sprawdza, czy do gniazda równoległego podłączona jest drukarka, i jaki jest jej aktualny status. Za pomocą różnych funkcji i programów pomocniczych można łatwo zlokalizować ewentualnie występujące problemy z drukarką, co ułatwi unięcenie błędów. Rozszerzona wersja programu z dysku 479. [Program.] Autor: Tom Kroener.

(Cdn.)

NAJLEPSZY ZNAWCA AMIGI

Postanowiliśmy znaleźć najlepszego znawcę Amigi w Polsce. Wiemy, że za takiego znawcę może uważać się każdy. Ale, niestety, sama wiedza to nie wszystko. Trzeba umieć ją jeszcze sprzedać, a przy tym mieć nieco szczęścia. Najprostszą metodą sprawdzenia jest test. Ogłaszamy zatem nasz wielki konkurs. A oto jego zasady:

➤ W konkursie może wziąć udział każdy Czytelnik Magazynu AMIGA, z wyjątkiem członków redakcji magazynu.

➤ Należy odpowiedzieć na wszystkie pytania zawarte w trzech kolejnych odcinkach konkursu — przez przekreślenie krzyżykiem kwadracika przy właściwej odpowiedzi.

➤ Wszystkie trzy wypełnione kupony konkursowe (oryginalna) należy przelać na adres Wydawnictwa Lupus w terminie do 15 grudnia 1992 (decyduje data stempla pocztowego) z dopiskiem "Konkurs - Amiga".

➤ Każdy Czytelnik może nadesłać dowolną ilość oryginalnych kuponów, z tym że w jednej kopercie może znaleźć się tylko jeden zestaw trzech kuponów konkursowych.

➤ Prenumeratorki Magazynu AMIGA (którzy zamówili już prenumeratę lub zrobią to do 30.10.1992) otrzymają gratisowo jeden dodatkowy zestaw kuponów konkursowych.

➤ Nagrodami w konkursie są: I nagroda — Amiga 600, II nagroda — dysk twardy A990, III nagroda — genlock. Ponadto rozlosujemy kilkanaście nagród poseszezeń, którymi będą dyskietki z programami i potrzebna prenumerata Magazynu AMIGA (jeżeli ta ostatnią nagrodę wylosuje osoba, która już jest prenumeratorem — oznaczać to będzie przedłużone prenumeraty).

➤ Losowanie nagród odbędzie się w dniu 21.12.1992. Wyniki zostaną podane w styczniowym numerze Magazynu AMIGA.

1) W jakim obszarze pamięci leżą wektory procesora MC68000?

- w obszarze, który wybiera sama Amiga podczas pracy w multitasking
- pomiędzy adresami \$0000 do \$03FF
- pod adresem \$DFF000

2) W jaki sposób można wyłączyć filtr dźwiękowy na wyjściu Audio w Amidze 500?

- wykorzystując rejestr hardware \$BFE001
- wyłącza się w taki sam sposób, jak dioda LED "POWER"
- Amiga 500 nie ma takiego filtra

3) Kto jest twórcą Pascala i Moduli 2?

- Niklaus Wirth
- Blaise Pascal
- Konrad Zuse

4) Co oznacza skrót ANSI?

- Amiga Numeric Serial Interface
- American National Standard Institute
- Amiga New Super Interface

5) Od jakiego nazwiska wzięta się nazwa "algorytm"?

- Jeff N. Algo
- Mohammad Ibn Musa al-Chwarizmi
- Giennadij Algorytowski

6) W jaki sposób można używać na Amidze adresów 24-bitowych?

- nie można
- przy Small Data Mode
- bez dodatkowych zabiegów

7) Ojcem języka C jest

- Dennis M. Ritchie
- Blaine Stroustrup
- Ken Thompson

8) Co stanie się ze zbiorami, których ikony za pomocą myszki przeniesiesz do szuflady "Trashcan"?

- zbiory zostaną zmasowane, a ich nazwy usunięte z katalogu, w którym się poprzednio znajdowały
- zbiory za pomocą specjalnego algorytmu zostaną "zamrożone" i nie będzie można ich odczytać
- zbiory zostaną skopiowane do specjalnego katalogu

9) Co dzieje się z danymi przekazanymi do urządzenia logicznego o nazwie NIL: (np. przez COPY nazwa danych >NIL...)?

- nic
- dane są przekoplowane do katalogu "Trashcan"
- system kopiuje te dane do katalogu o nazwie NIL

10) Dlaczego część pamięci Amigi nazywa się "FAST-Ram"?

- mikroprocesor Amigi ma szybszy dostęp do tej pamięci
- pamięć ta zbudowana jest z szybszych układów pamięci
- przez pomyłkę zainstalowano tam niewłaściwe układy

11) Co się stanie, gdy po wpisaniu: "date 1-Jan-78" naciśniesz [RETURN]?

- Amiga zawiesi się
- aktualną datą systemową stanie się 1 stycznia 1978
- nic się nie stanie

12) W jaki sposób można przerwać wykonującą się sekwencję startową?

- klawiszem [Esc]
- klawiszami [CTRL] + [D]
- klawiszami [CTRL] + [X]

13) Kto jest "ojcem" Amigi (i tzw. GURU)?

- Jeff Minter
- Jay Miner
- Joy Minner

14) Ile wersji Amigi wyprodukowała firma Commodore?

- 11
- 12
- 13

15) Genlock służy do

- dekodowania programów TVSAT za pomocą Amigi
- nakładania animacji Amigi na film wideo i zapisu wyniku na dyskietkę
- mieszania obrazu ze źródła wideo i z Amigi i zapisu wyniku na taśmę wideo

16) Do 1 lipca 1992 Fred Fish puścił w obieg:

- 546 dyskietek Public Domain
- 646 dyskietek Public Domain
- 746 dyskietek Public Domain

17) Ile (szacunkowo) Amig jest używanych w Polsce?

- 5000
- 50000
- 200000

18) W przypadku, gdy zepsuje Ci się mysz, to:

- nie możesz uruchomić Amigi
- możesz korzystać tylko z określonych programów
- możesz korzystać ze wszystkich programów, choć nie zawsze będzie to wygodne

Oceny, pochwały, propozycje

(...) Teraz o nowym piśmie. Mam nadzieję, że będzie to właśnie to, na co czekaliśmy. Początek jest interesujący. Nareszcie zapowiada się pismo na przyzwoitym poziomie (mam nadzieję, że cena będzie również przyzwoita) wyłącznie o Amidze. No cóż, C&A to nie tylko Amiga, a mało który amigowiec jest jeszcze zainteresowany C64, więc połowa pisma jest nie dla niego. Zawartości nie skomentuję. Pozostałe pisma (64+4, Amigowiec, Kebab, Amiger) są właściwie amatorskie i stosunek ich zawartości do ceny jest niezbyt korzystny. Wy macie największe szanse na zostanie Nr 1 dla Amigi w Polsce. Fakt posiadania zaplecza w postaci artykułów zagranicznych bardzo Wam pomoże, ale powinniście również zamieszczać artykuły polskie. Artykuły zagraniczne w większości wypadków są promocją różnych produktów najczęściej w Polsce niedostępnych, a jeśli dostępnych to tylko dla niewielu Polaków. Można je traktować jako ciekawostkę (co słyhać w świecie), ale nie mogą stanowić całości. Kilka stron o grach jest konieczne (trzeba czymś przyciągnąć szarego użytkownika Amigi), ale umiarkowanie. Dobrym pomysłem byłoby zainteresowanie czytelników poważniejszymi zastosowaniami Amigi, nie tylko grami. (...)

Bardzo nam się podobał artykuł o Fredzie Fishu. My również jesteśmy zafascynowani jego biblioteką i staramy się przybliżyć ją polskim użytkownikom, jak na razie z dość dobrym skutkiem. ("Dochodzę do wniosku, że przed przeczytaniem Waszego ogłoszenia w jednym z ubiegłorocznych numerów 64+4 wykorzystywałem powyżej 20% możliwości Amigi (nie mówiąc o moim mózgu). Od tamtego czasu twierdzą, że >Mysz Shareware< zrobiła więcej dla tzw. komputeryzacji niż wszystkie nasze czasopisma komputerowe łącznie. Każdy, kto twierdzi inaczej, kończy z przestrzelonym procesorem." — cytat z jednego z listów.) Dobrym pomysłem jest zamieszczanie polskich opisów zawartości dysków Fisha (choć my zaczęlibyśmy gdzieś od numeru 400 — dużo ciekawych, nieprzestarzałych programów, sporo animacji, np. Schwartz). (...)

Krótko o nas i naszych celach. Chcemy stworzyć kompletną bibliotekę programów public domain i shareware, w której każdy mógłby znaleźć program, jakiego szuka (...). Za punkt wyjściowy przyjęliśmy kolekcję Freda Fisha (...). Dodatkowo myślimy o wprowadzeniu innych liczących się kolekcji (...). Ponadto stworzyliśmy własną kolekcję. Pod własną firmą rozpowszechniamy dyskietki stworzone przez nas (jak Mysz 022; 001-010; 031-037), dyskietki różnych autorów (grup) nie należące do bibliotek (np. megadema) oraz pojedyncze dyskietki z różnych bibliotek nie noszące numerów (Mysz 038, 039). (...)

Robert Jacaszek
"Mysz Shareware"
Gdynia

List jest bardzo obszerny (nawet po skrótach), więc po kolei.

1. Nie będę komentował zawartych w liście ocen pism funkcjonujących na rynku Amigi (osobiście oceniam je wyżej). Podobnie jak autor listu, także my mamy nadzieję, że z czasem Magazyn AMIGA wejdzie do czołówki polskich pism "amiganckich". W każdym razie, będziemy nad tym usilnie pracować.

2. Co do tłumaczeń artykułów: chcielibyśmy, aby pod koniec roku 80% tekstów ukazujących się w naszym miesięczniku stanowiły artykuły oryginalne, a jedynie 20% tłu-

maczenia. Liczymy tu także na Waszą aktywność pisarską.

3. Gry. Staramy się nie zapominać ani o graczach (granie nie wydaje nam się niczym naganym), ani o użytkownikach zainteresowanych poważnymi zastosowaniami swego komputera. Kwestia proporcji jest zawsze otwarta i zależy od Waszych sugestii.

4. Shareware. Działalności "Myszy Shareware" można jedynie przyklasnąć. Być może również redakcja Magazynu AMIGA wniesie coś do tej sprawy. Póki co, staramy się koncentrować na wydawaniu pisma, gdy jednak dojrzejemy organizacyjnie — kto wie...

Czy MA będzie dostępny?

(...) Gdy kupiłem lipcowy numer wydawanego przez Was popularnego czasopisma, jakim jest "ENTER", z wielkim zadowoleniem przeczytałem wiadomość, że od września będzie Wasz nowy produkt, którym jest "Magazyn AMIGA". Bardzo mnie to cieszy. Jednak z zadowoleniem "przyszedł" także kłopot. Mianowicie nie wiem, czy "Magazyn AMIGA" będzie dostępny w całym kraju. Bardzo bym chciał, żeby to czasopismo można było kupić w każdym kiosku (nawet we wsiach). (...)

Dominik Biernacki
Warta woj. sieradzkiej

Pierwszy (wrześniowy) numer Magazynu AMIGA został wydrukowany w 45 tysiącach egzemplarzy, z czego 40 tysięcy trafiło do RUCH-u (reszta to prenumerata oraz inne kanały dystrybucji). Sądzę więc, że pismo jest dosyć łatwo dostępne w kioskach.

Jeśli jednak będziecie mieli kłopoty z kupieniem naszego miesięcznika w jakiejś miejscowości, to prosimy o poinformowanie nas (spróbujemy wpłynąć na tzw. rozdzielnik i zwiększyć przydzielaną do Waszej miejscowości liczbę egzemplarzy).

AMIGO — do szkoły!

(...) Jestem nauczycielem matematyki w szkole podstawowej. (...) Kupiłem ten komputer (Amigę — przyp. red.) głównie z myślą o moich dzieciach (chłopcy 7 i 9), chociaż doradzano mi XT, a znajomy programista z firmy komputerowej przez dłuższy czas nie mógł przeboleć, że wyrzuciłem pieniądze. Uważam, że jednym z celów nowego magazynu powinno być zdjęcie z Amigi odium maszyny wyłącznie do gier. Marzy mi się, aby ten komputer trafił do szkół. Niestety, np. dyrektor mojej szkoły wspominał, że chciałby kupić jakiś dobry komputer. Gdy zaproponowałem Amigę, zachnął się — "za tę cenę możemy mieć PROFESJONALNY (!!!) XT". To nieporozumienie, przecież Amiga ze względu na relacje cenowo-jakościowe, wszechstronność, możliwość wprowadzenia polskiej wersji Workbench, jest komputerem idealnym do celów dydaktycznych. Mam więc nadzieję, że w nowym piśmie znajdzie się stały dział poświęcony dydaktycznym zastosowaniom Amigi. (...)

Co jeszcze chciałbym znaleźć? — informacje na temat:

- możliwości współpracy Amigi z domowymi, elektronicznymi instrumentami muzycznymi przez MIDI (opis konkretnych programów),
- współpracy z modemem (opis dostępnego u nas sprzętu wraz z cenami),
- możliwości korzystania z Amigi jako faksu,
- rozbudowy 500-ki z opisem możliwości zastosowań. (...)

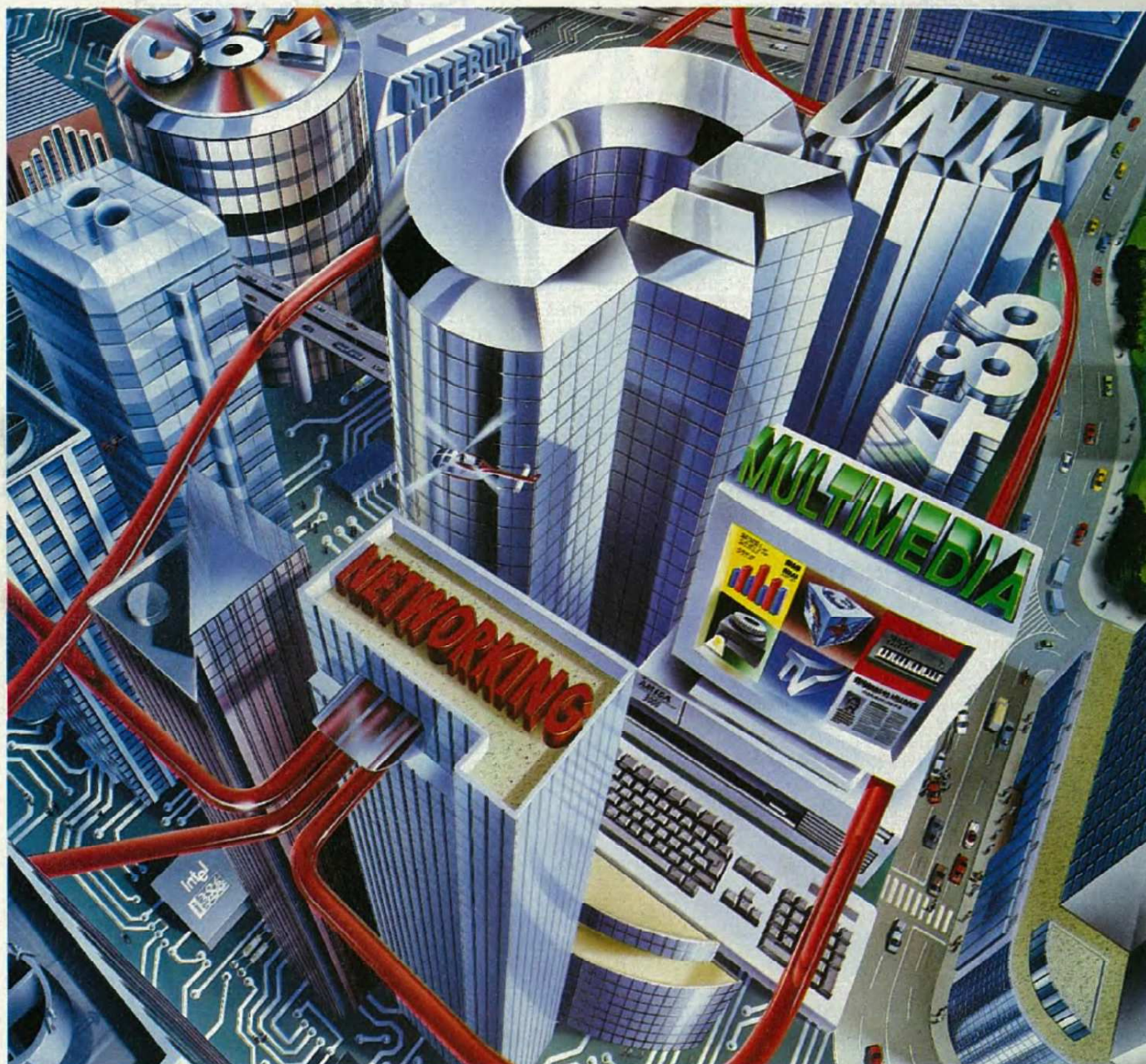
Nawracanie jest trudną sztuką, zwłaszcza gdy nawracany nie chce słuchać. Wydaje się, że przykładów podobnych do opisanego w liście możecie przytoczyć wiele; sądzą jednak, że nie warto — sami siebie przekonywać musimy, a ci, których chcielibyśmy przekonać, słów tych nie przeczytają. Warto natomiast pokazać przykłady edukacyjnych zastosowań AMIGI. Jeśli znajdziecie takie, gorąco prosimy o ich przedstawienie — dostarczymy w ten sposób argumentów apostołom.

Jeśli chodzi o inne wymienione tematy, zapisałyśmy je w kajecie — czekają na realizację. Mamy w związku z tym propozycję: spróbujemy uruchomić rubrykę "Artykuł na zamówienie", w której publikować będziemy materiały na tematy wskazane przez Czytelników. Oczekujemy listów z propozycjami najważniejszych i najciekawszych Waszym zdaniem tematów.

Listy wybrał i w imieniu redakcji na nie odpowiedział

Kolega Redaktor

COMMODORE-KONCERN TECHNOLOGII



PION KOMPUTERÓW OSOBISTYCH

KOMPUTERY Z
SYSTEMEM
OPERACYJNYM
MS-DOS,
KOMPUTERY TYPU
NOTEBOOK

PION SIECI KOMPUTEROWYCH

NOVELL
UNIX
LAN, WAN

PION KOMPUTERÓW AMIGA

AMIGA 3000
AMIGA 2000
MULTIMEDIA

PION KOMPUTERÓW DOMOWYCH

CDTV®
AMIGA 500
AMIGA 600
C 64



Commodore