



Z MIKROKOMPUTEREM NA TY

NR INDEKSU 353965  
PL ISSN 0860-1674

# Bajtek

MAGAZYN KOMPUTEROWY

NR 4(57-58)'90 CENA 5000

## KONKURS!



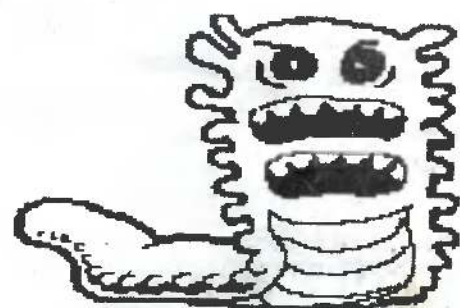
PIERWSZE CHWILE  
Z KOMPUTEREM

MAGA  
ZYNIER

PIĘKNO MATEMATYKI

PROCEDURY SYSTEMOWE  
AMSTRADA

INDIANA JONES



wirusy...

# PO RAZ CZWARTY ...

Spotykając się z Państwem po raz czwarty w tym roku mamy kilka miesięcy ciężkiej i wytężonej pracy za sobą. Spójrzmy jakie są jej rezultaty w tym numerze.

Wychodząc naprzeciw postulatam Czytelników, traktujących nasze pismo jako lekturę podstawową przy pierwszym kontakcie z komputerem, w każdym klanie umieściliśmy krótki materiał poświęcony właśnie temu zagadnieniu.

Przywiązując wielką wagę do edukacyjnych zastosowań komputera, postanowiliśmy od nowego roku wprowadzić nowy klan w Bajtku, o roboczej nazwie — „Po dzwonku”. Krótkim wprowadzeniem do tej tematyki jest materiał pod tytułem „Po co belfrowi komputer”.

Z kolei w klanie Spectrum znajdują Państwo m.in. listing gry Magazynier, publikowany wcześniej na łamach Bajtka w wersji na małe Atari. Dla posiadaczy stacji Timex FDD 3000 jest do wstukania zestaw procedur graficznych Turbo Pascala.

Umieszczony w Klanie Amstrad opis programu do analizy syntaktycznej wyrażeń algebraicznych przyda się na lekcjach matematyki. Osoby korzystające z CPC 6128 znajdą gotowy program umożliwiający wydruk ekranu, a właściciele wszystkich komputerów serii CPC zadowolą zapewne wznowienie cyklu opisywanego procedury systemowe.

Akcentem świątecznym jest konkurs, którego rozwiązanie daje szansę wylosowania komputera. Mamy nadzieję, że temat nie jest tak trudny, jak poprzednio, i że otrzymamy więcej poprawnych odpowiedzi niż rok temu.

Klan IBM, w tym numerze dość monotematyczny, poświęcony jest głównie wirusom i sposobom, w jaki należy ich unikać. Istotną częścią klanu jest krótki opis programu antywirusowego MkS.Vir, udostępnionego nam przez pana Marka Sella.

Mamy nadzieję, że testy dobrych i użytecznych programów oraz dostępnego sprzętu, będą naszą stałą praktyką. Aktualnie w kolejce czeka program Kat90 będący dla wielu użytkowników rozwiązaniem dręczącego ich problemu

utrzymania porządku w zbiorach znajdujących się na więcej niż kilkunastu dyskietkach.

Otrzymaliśmy również trochę sprzętu do testowania. Po prawie dwóch latach jednostronnej korespondencji z zakładami Mera Poltik i Mera Błonie dotarli do redakcji: ploter i dwie drukarki. Firma Biazet przekazała nam popularny monitor i tzw. telemonitor, pan Marek Tokarski zarzucił nas kilkunastoma joystickami produkowanymi w niedalekich Chinach. Testy tych wszystkich urządzeń wypełnią następane numery Bajtka.

Zdajemy sobie jednak sprawę z pewnego kontrastu jaki panuje między rzeczami, wokół których ogniskują się nasze zainteresowania, a tym co jest dostępne w innych krajach. W Stanach Zjednoczonych firma IBM wprowadziła nową serię komputerów domowych typu PS/1. Model podstawowy to AT 286; śladem wielu innych producentów wycofano z oferty sprzęt oparty o procesor 8086. W Wielkiej Brytanii, pod hasłem „Mac dla mas”, zalew sprzętu domowego firmy Apple. Najtańszy model — Macintosh Classic — kosztuje poniżej 600 funtów. Z kolei pan Alan Sugar oferuje nową serię komputerów Amstrad 3000.

Za wcześnie jeszcze, aby każdy komputer pracował w systemie UNIX, ale już w tej chwili japońska firma Toshiba lansuje przenośnego laptopa T5100 wyposażonego w ten system operacyjny. Ciekawą ofertę ma też Hewlett Packard. Najnowsza drukarka strumieniowa DeskJet 500 w cenie kilkuset funtów zapewnia jakość podobną do tej jaką mamy w drukarkach laserowych, a jest znacznie tańsza w eksploatacji.

Wracając na ziemię — powoli finalizują się sprawy związane z przekazaniem nam przez Komisję Likwidacyjną tytułu Bajtek. Mamy również dane z kolportażu o zwrotach za numer 5—6/90. W sumie są to informacje pozwalające z nadzieją patrzeć w przyszłość. W tym też nastroju składamy Państwu najserdeczniejsze życzenia Wesołych Świąt i Szczęśliwego Nowego Roku.

Jarosław Młodzki

## PRENUMERATA

Przez cały czas przyjmujemy zamówienia na prenumeratę wszystkich trzech wydawanych przez nas tytułów. Prenumeratę można zamówić na dowolny okres, na dowolną liczbę egzemplarzy dowolnych tytułów. W tym celu należy wpłacić odpowiednią kwotę na nasze konto:

**Spółdzielnia „Bajtek”**  
Bank „Agrobank S.A.”  
479994-1834-131  
ul. Grochowska 262  
04-398 Warszawa

Wstępny koszt rocznej prenumeraty jednego egzemplarza Bajtka wynosi 66000 zł, Top Secret (6 numerów) 33000 zł, Mojego Atari (6 numerów) 30000 zł. Jeśli cena któregoś z tytułów zostanie

podniesiona, wpłata będzie potraktowana jako przedpłata, i potrzebna będzie dopłata. O jej konieczności będziemy w tym miejscu informować. Prosimy o bardzo staranne wypełnianie przekazów długopisem drukowanymi literami, i podawanie na odwrocie wszystkich trzech odcinków informacji o tym, jakiego okresu i których tytułów dotyczy prenumerata. Niewyraźnie lub błędnie wypełnione przekazy mogą być źródłem opóźnień i błędów w realizacji zamówień.

Uwaga: po otrzymaniu przekazu wysyłamy pocztą najwcześniejszy możliwy numer. Ilość wystawionych numerów będzie zgodna z zamówieniem. W razie błędnej realizacji prenumeraty, błędu w adresie, lub nieotrzymania pierwszego numeru zamówionych pism w ciągu dwóch tygodni od momentu ukazania się ich w kioskach, prosimy o kontakt w celu wyjaśnienia sprawy.

# Bajtek

MAGAZYN KOMPUSEROWY

**Redaguje Kolegium:**  
Marcin Borkowski, Marek Czarkowski, Łukasz Czekajewski, Janusz Jarmoch, Jarosław Młodzki (red. nacz.), Waldemar Nowak, Maciej Pietraś, Marcin Przasnyski, Wanda Roszkowska, Wojciech Zientara

**Szefowie Klanów:**  
Amstrad — Jonasz Mayer  
Atari — Wojciech Zientara  
Commodore — Klaudiusz Dybowski  
Co Jest Grane — Łukasz Czekajewski  
IBM — Marcin Borkowski  
Micro Magazyn — Janusz Jarmoch  
Spectrum — Maciej Pietraś

**Andrzej Pilaszek**  
**Mieczysław Płacheta**  
**Marek Sawicki**  
**Piotr Sumara**  
**Michał Szokoło**  
**Tomasz Tarczyński**  
**Stanisław Winiecki**

**Opracowanie graficzne:**  
Wanda Roszkowska

**Redaktor techniczny:** Maria Radziwińska  
**Zdjęcia:** Leopold Dzikowski

**Wydawca:**  
Spółdzielnia „Bajtek”  
ul. Wspólna 61  
00-687 Warszawa

**Skład i Druk:**  
Prasowe Zakłady Graficzne w Ciechanowie  
**Fotoskład:** Grażyna Kurzątkowska  
**Montaż:** Grażyna Ostaszewska  
**Korekta:** Maria Krajewska  
Teresa Rutkowska  
**Nakład:** 105 tys. egz. Zam. 66820

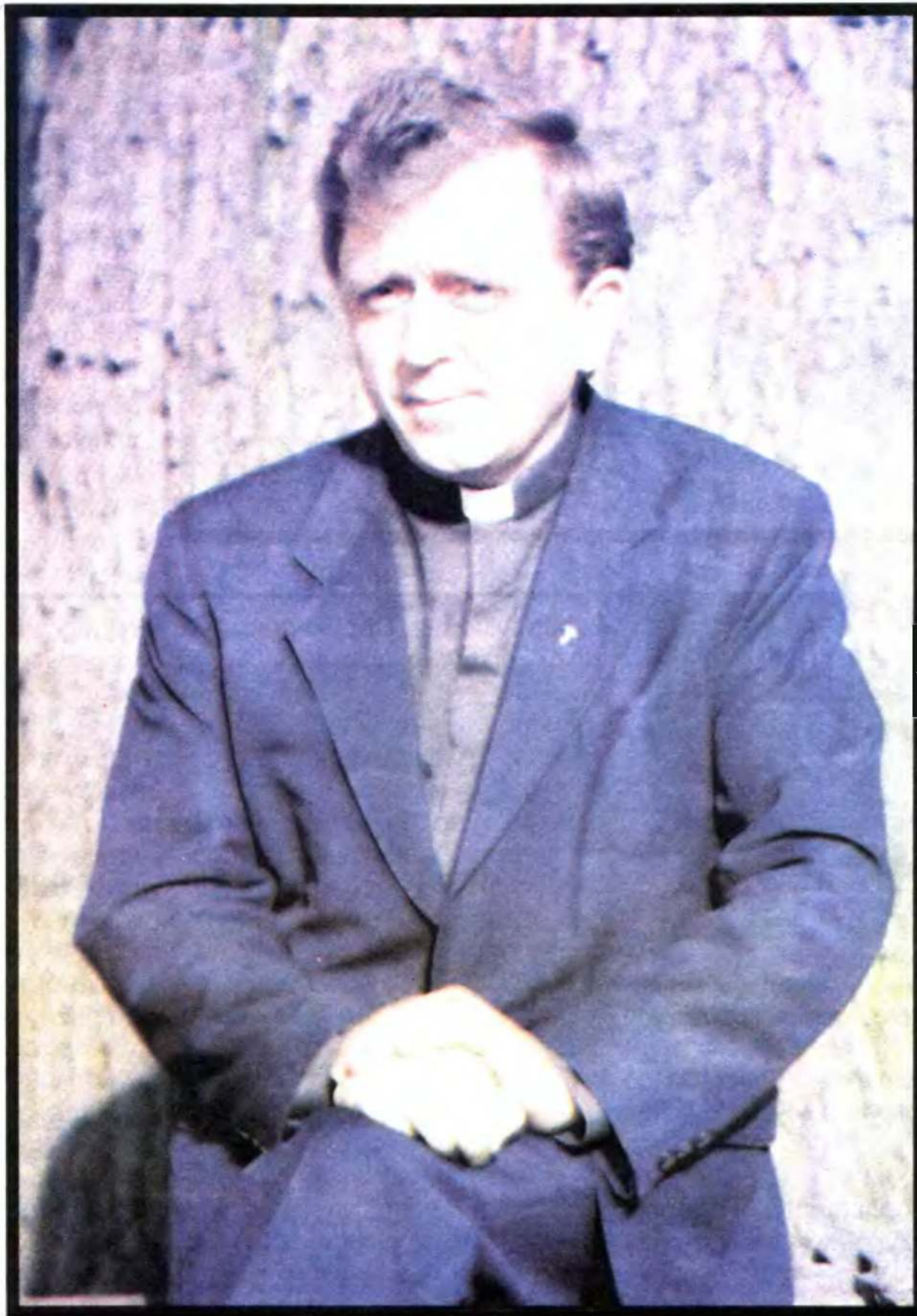
**Redakcja:**  
ul. Wspólna 61  
00-687 Warszawa  
Tel. 21-12-05

**Stali współpracownicy:**  
Marcin Bójko  
Jarosław Burczyński  
Piotr Kos  
Robert Magdziak  
Grzegorz Ostapiuk



<b>W OBLICZU NIESKOŃCZONOŚCI . . . . .</b>	<b>3</b>
<b>MICROMAGAZYN . . . . .</b>	<b>4</b>
<b>JĘZYK „C”</b>	
<b>DLA NAJMŁODSZYCH . . . . .</b>	<b>6</b>
<b>PO CO BELFROWI KOMPUTER? . . . . .</b>	<b>7</b>
<b>KLAN ATARI . . . . .</b>	<b>8</b>
: AST — drugie spojrzenie	
: Przerwania w Action!	
: Jeszcze o sortowaniu	
: Piękno matematyki	
: Szkoła przetrwania	
: Licznik czasu	
: Zapis i odczyt rysunków	
: RamBrandta	
<b>KLAN COMMODORE . . . . .</b>	<b>12</b>
: Pierwsze kroki z Commodore	
: Just for FUN!	
<b>KONKURS . . . . .</b>	<b>13</b>
<b>KLAN SPECTRUM . . . . .</b>	<b>14</b>
: Pierwsze kroki ze Spectrum	
: Balistyki	
: Timex 80 KB	
: Magazynier	
: Turbo Pascal — procedury graficzne	
: CAT	
: Język Maszynowy cz.3	
<b>ROZWIĄZUJEMY KONKURS 19 CO JEST GRANE? . . . . .</b>	<b>20</b>
: Indiana Jones III	
: Piłka w grze	
: Kick Off 2	
: SOS	
<b>KLAN AMSTRAD . . . . .</b>	<b>24</b>
: Analiza wyrażeń w Turbo Pascalu	
: Tylko dla początkujących	
: Procedury systemowe	
: Amstrada	
: Wydruk ekranu na CPC 6128	
<b>KLAN IBM . . . . .</b>	<b>28</b>
: Wirusy...	
: MkS vir	
: Ochrona dysków twardych przed wirusami	
: Oswajanie PC-eta	
: Strzeż się wirusów	
: Yankee Doodle	
<b>ILE DANYCH POTRZEBA NA JEDNĄ BAZĘ . . . . .</b>	<b>32</b>
<b>GIEŁDA, IBD . . . . .</b>	<b>34</b>
<b>DROGI BAJTKU . . . . .</b>	<b>35</b>
<b>WSZYSTKO DLA WSZYSTKICH . . . . .</b>	<b>36</b>
<b>ORKIESTRA TO CZY MASZYNA? . . . . .</b>	<b>40</b>

Rozmowa  
z księdzem  
Janem  
Chrapkiem,  
wykładowcą  
Uniwersytetu  
Warszawskiego  
i Uniwersytetu  
Gregoriańskiego  
w Rzymie



*Człowiek jest tylko trzcina,  
najwęższą w przyrodzie,  
ale trzcina myślącą.*  
Blaise Pascal

**Proszę Księdza, czy zmiany, jakie niesie ze sobą masowe zastosowanie komputerów w życiu społeczeństw, nie zagrażają człowiekowi?**

Od wielu lat prowadzone są badania nad społecznymi skutkami rewolucji technologicznej, jaka dokonuje się na naszych oczach. Świadectwem tego jest opublikowany już w roku 1982 głośny raport Klubu Rzymskiego pt. „Mikroelektronika i społeczeństwo”. Kościół także nie uchyla się od refleksji nad tym zjawiskiem, choć przyznam, że jest jeszcze za wcześnie, aby angażował cały swój autorytet w jego ocenę.

Osobiście zajmuję się środkami masowego przekazu, w których pracy coraz większą rolę odgrywają komputery. Trudno dziś bez nich wyobrazić sobie sprawny pracę agencji prasowych, stacji telewizyjnych czy redakcji gazet. W tym sensie oczywiście spełniają one pożyteczną rolę. Czy jednak zagrażają?

Każda nowa technologia wymaga nowej wrażliwości osobowej i etycznej. Komputer, jak każde inne narzędzie, może być użyty także przeciw człowiekowi.

**Niestety coraz częściej dostrzegamy przykłady zjawisk mogących zagrazić tak jednostce, jak i całym grupom społecznym. Mam na myśli ogromną koncentrację informacji, a w konsekwencji możliwość manipulacji.**

Wszystko zależne jest od człowieka, nie od maszyny. Często nie zdajemy sobie sprawy, jak wielki wpływ na życie społeczne i ekonomiczne, struktury polityczne, a nawet osobowość, mają nowe technologie.

Zjawiska te odnoszące się do techniki komunika-

cji masowej badali między innymi Harold Innis, kanadyjski uczyony i ekonomista, Walter Ong — jezuita, profesor uniwersytetu w Saint Louis, czy Marshall Herbert MacLuhan. Zastanawiali się, w jaki sposób sama natura techniki zmienia sposób patrzenia na rzeczywistość. Myślę, że podobnie rzecz się ma z komputerami. Nie dysponujemy jednak dziś dokładnymi wynikami badań na ten temat.

**Sądzę, że Ksiądz jako kapłan zobowiązany jest do refleksji nad kondycją człowieka w zetknięciu z nowoczesną techniką.**

To oczywiście. W centrum zainteresowania Kościoła zawsze stał człowiek i jego problemy. Zawansowane technologie, poza niewątpliwymi korzyściami, niosą także zagrożenia. Takie jak nadmierna fascynacja techniką, w której stajemy się już tylko dodatkiem do maszyny.

Być może zabrzmi to paradoksalnie, ale sądzę, że wkrótce staniemy się świadkami gwałtownego zwrotu w stronę humanistyki. Obserwuję coraz większe zmęczenie ludzi zagubionych we współczesnej cywilizacji. Mam nadzieję, że wielu zwróci się w swych poszukiwaniach w stronę Boga.

Nauka i technika nie są wcale tak odległe od religii.

**Czy ma to także związek z komputerami?**

Bezpośrednio nie, ale czy wie pan, że wielu

uczonych zajmujących się poważnymi problemami naukowymi to ludzie głęboko religijni. Jeśli porównać człowieka do samotnej wyspy otoczonej oceanem NIEZNANEGO, jakże możemy się dziwić, że odkrywa on istnienie nieskończoności. W tym miejscu dotykamy prawd ostatecznych. Kończy się poznanie w dających się opisać kategoriach, a zaczyna WIARA.

I nie ma znaczenia, czy zagłębiamy się w kosmos gwiazd czy atomów.

Jestem przekonany, że komputer dla wielu młodych ludzi na początku jest tajemniczą skrzynką z klawiszami. Muszą ją zbadać i odkryć dla siebie. Często z ogromnym wysiłkiem, ale na tym polega proces POZNANIA, na którego końcu — w co wierzę — jest Bóg.

**Jest także wielu myślących inaczej..**

To oczywiście. Starłem się podzielić swymi przemyśleniami.

**Jestem ciekaw, czy Kościół jako instytucja także korzysta z komputerów?**

W warszawskiej Akademii Teologii Katolickiej w badaniach nad Biblią profesor Mendala stosuje techniki obliczeniowe. Także wiele instytucji watykańskich korzysta z komputerów, na przykład słynna Biblioteka Watykańska jest obecnie komputeryzowana. Nie wiem, jaka firma się tym zajmuje, jak sądzę ze względu na reklamę, jest to tajemnica. To ogromne przedsięwzięcie. Biblioteka należy do najstarszych i największych na świecie.

**Czy znaczy to, że kształcą się księża — programistów?**

Na razie nie, ale wśród księży i zakonników są też absolwenci fizyki, matematyki i uczelni technicznych, którzy doskonale znają języki programowania i potrafią wykorzystać swoją wiedzę. W przyszłości być może Kościół będzie zmuszony kształcić własnych fachowców.

**Mam wrażenie, że będzie to prawdziwa rewolucja.**

To trudno orzec. Wszyscy musimy dostosować się do zmian zachodzących w świecie, ale kapłani są szczególnie zobowiązani do ostrożności. Proszę mnie dobrze zrozumieć. Nikomu nie wolno bezkrytycznie przyjmować najbardziej nawet atrakcyjnych nowości.

Potrzebny jest czas i dystans w ocenie. Wiemy, że techniki komputerowe mogą być bardzo użyteczne, ale nie wiemy, jakie powstaną zagrożenia.

Jak pogodzić ewangeliczny warunek ubóstwa z kosztownym urządzeniem parafii?

Czy nie obawia się pan, że i w naszym kraju zapomnimy o ubogich? Są młodzi ludzie dysponujący nowoczesnym sprzętem komputerowym, a obok ich koledzy nie mogą nawet o tym marzyć. Powinniśmy o tym pamiętać.

**Zaczelśmy naszą rozmowę cytatem wielkiego francuskiego myśliciela Blaise Pascala, który był człowiekiem głębokiej wiary...**

...i wybitnym naukowcem. Jego „Myśli”, z których pochodzi ów fragment, stanowią jedno z największych osiągnięć filozofii europejskiej. Niklaus Wirth złożył piękny hołd wielkiemu Francuzowi, nazywając swój język programowania jego imieniem.

Rozmawialiśmy o Kościele, wierze, komputerach i poznaniu. Niech na zakończenie naszej rozmowy wolno mi będzie zacytować Pascala:

*Wiekuista cisza tych nieskończonych przestrzeni przeraża mnie.*

Marek Czarkowski

# W OBLICZU NIESKOŃCZONOŚCI

# ZŁOM KOMPUTEROWY

Zdaniem specjalistów od ochrony środowiska Europie Zachodniej zagraża w nadchodzących latach zalew starych terminali komputerowych i wyeksploatowanych PC. Groźba ta zmusza producentów do opracowania programu utylizacji niepotrzebnych urządzeń.

Złom komputerowy zawiera stal, tworzywa sztuczne, oleje, smary, baterie, gazy i luminofory z monitorów. Wszystko to tworzy ekologicznie wybuchową mieszankę. Niektóre substancje wchodzące w skład elementów komputera uwalniają podczas spalania bardzo szkodliwe związki chemiczne. Ale odpady te nie muszą nas zatrwać. Ze względu na możliwość ponownego wykorzystania stanowią one atrakcyjny surowiec wtórny.

Obroncy środowiska naturalnego domagają się ustawowego zobowiązania producentów do usuwania i utylizacji wyeksploatowanych urządzeń. Jest tego dosyć dużo. Według oficjalnych danych w RFN sprzedano dotychczas ponad 11.000.000 komputerów osobistych i różnych systemów komputerowych. Eksperti oceniają, że powoduje to powstawanie około 7.500 ton złomu rocznie. Prognoza na przyszłość przewiduje coroczny przyrost ilości odpadów o 5 do 10 procent.

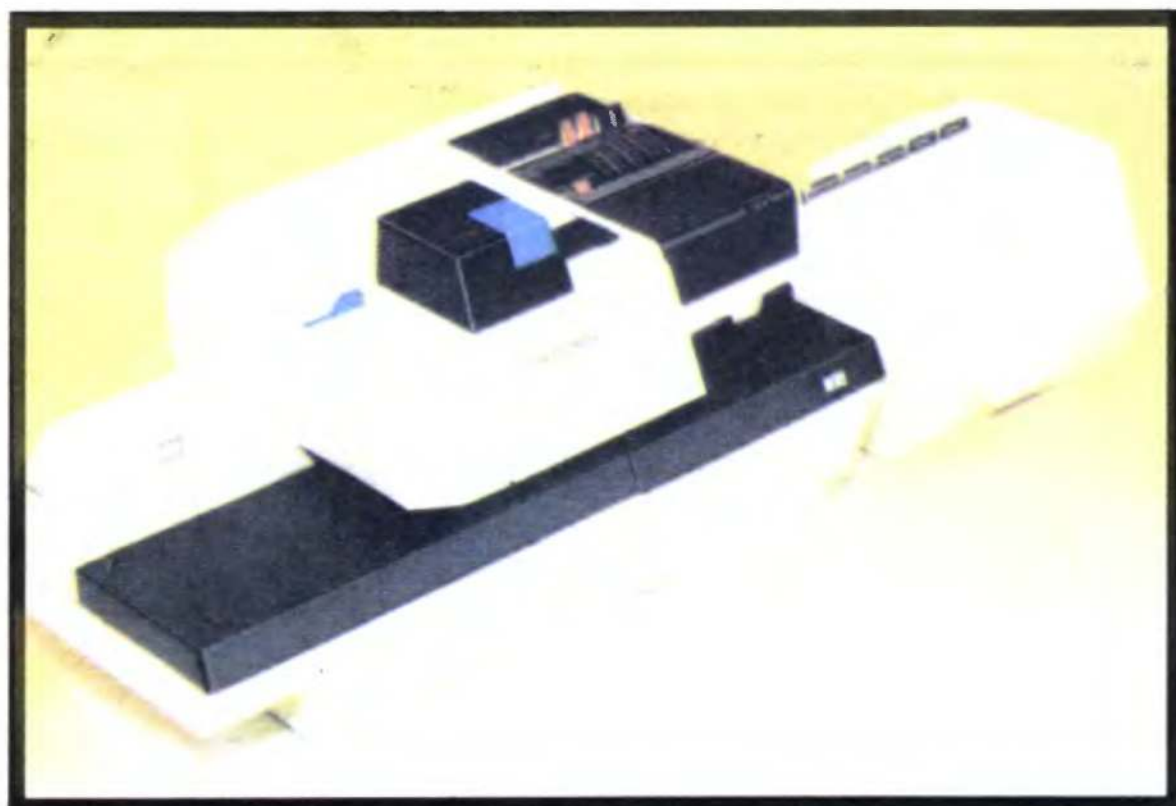
Komputerowy potentat IBM, chcąc pozostać w zgodzie z wymogami ochrony środowiska, opracował koncepcję odbioru od użytkownika, utylizacji i ponownego wykorzystania niektórych surowców wchodzących w skład złomu. Na razie oferta ta dotyczy wyłącznie własnych wyrobów IBM. Na ogół przynosi to producentowi dodatkową korzyść. Klient który pozbywa się starego sprzętu, kupuje zazwyczaj nowsze i doskonalsze modele. W ślad za IBM poszły również i inne firmy. Niektórzy jednak uchylają się przed tym wskazując, że znacznie większym zagrożeniem jest elektronika konsumpcyjna. Na śmietniki w RFN ma przecież trafiać rocznie ponad 100.000 ton starych odbiorników radiowych i telewizyjnych.

(JJ)

## DOPALACZ DO PCW

Brytyjska firma Cirtech, znana z produkcji dysków twardej do komputerów Amstrad PCW, wprowadziła na rynek jesienią tego roku nowe urządzenie o nazwie Sprinter. Jest to przystawka do komputera PCW zawierająca procesor Z80 taktowany zegarem 10MHz. Jej podstawową zaletą jest prawie trzykrotne przyspieszenie wszystkich operacji wykonywanych przez komputer. Dodatkowo możliwe jest zwiększenie pamięci RAM o 256 KB lub 1 MB. Cena urządzenia w wersji podstawowej wynosi 78 funtów, a każde 256 KB dodatkowej pamięci kosztuje 30 funtów.

(JM)



# FRANKO

Co prawda nie jest to komputer, ale całe urządzenie sterowane jest mikroprocesorem. Frankownica angielskiej firmy Pitney Bowes służy do stemplowania (dawniej mówiło się „frankowania”) listów. Maszyna znajduje zastosowanie w dużych biurach i instytucjach, które po podpisaniu odpowiedniej umowy z Poczta mogą w własnym zakresie stemplować listy.

W Polsce jest to jeszcze rzadkość, ale na świecie wiele nawet małych firm korzysta z tego rodzaju urządzeń. Mi-

## ELEKTRONIKA W TOTO-LOTKU

Towarzystwa zajmujące się organizacją loterii i gier liczbowych na terenie RFN starają się przyciągnąć jak najwięcej klientów do punktów przyjmujących zakłady. Ułatwieniu gry w totolotka lub na loterii pieniężnej służą długoterminowe abonamenty. Podobnie jak w przypadku prenumeraty czasopism wystarczy wypełnić

kupon tylko raz. Dopóki gracz nie złoży pisemnej rezygnacji, nie musi się w ogóle fatygować do kolektury. Należności za kolejne zakłady są pobierane automatycznie z jego konta.

System ten nie wystarczy organizatorom. Zamierzają oni udoskonalić metody sięgania do portfeli grających. Tygodnik „Stern” podał informację o opracowywanym właśnie projekcie wyposażenia punktów totolotka w terminale komputerowe. Jeśli plan zostanie urzeczywistniony, składane kupony będą w pełni automatycznie sprawdzane przez elektroniczne urządzenia rejestrujące. Wszystkie dane zostaną z szybkością błyskawicy przekazane do centralnego komputera towarzystwa lo-

teryjnego. System ten umożliwi przeprowadzanie losowań szczęśliwych numerów nawet codziennie.

Elektroniczne wyposażenie ma kosztować aż ćwierć miliarda marek. Ale pieniądze szybko się zwrócą. Do kieszeni graczy powraca w formie wygranych tylko około 50% kwoty, która wpływa na zakłady. Z drugiej połowy zostaje organizatorom wystarczająco dużo, nawet po odprowadzeniu lwiej części w formie podatków do kasy państwowej.

Niemcy z RFN wydają na hazard rocznie około 20 mld. marek. Nic dziwnego, że organizatorzy totolotków chcą uszczknąć przy pomocy elektroniki jak największą część tej sumy. (JJ)

## Nowe komputery wujka Alana



Już od dłuższego czasu chodziły plotki o przygotowaniu do produkcji komputera AMSTRAD PLUS. Wiść gminna głosiła, że będzie to zupełnie nowa jakość. Szepetano coś o nowym sterowniku graficznym, sześciu kanałach dźwięku, karcie EPROM-ów, a nawet o zmianie procesora z Z-80 na Motorola 68000! Mimo to nowe komputery są zaskoczeniem dla (prawie) wszystkich, nie dla każdego miłym. Pierwsza w szeregu jest konsola (urządzenie tylko i wyłącznie do grania) o poetycznej nazwie GX4000. Następnie AMSTRAD 464 PLUS i 6128 PLUS. Co oznacza ten PLUS? Oznacza on, że jako 6128 PLUS dostajemy przekonstruowanego CPC 6128 wraz z konsolą w jednej obudowie. Podobnie 464 PLUS, chociaż w tym przypadku mamy magnetofon zamiast dysków. Oba w prawie identycznych obudowach (w 464 magnetofon nad klawiaturą, w 6128 dysk wsuwany z boku), z klawiaturą wzorowaną na CPC 6128 i dziurą na cartridge (po naszymu: magazynek). Określenie „przekonstruowany” oznacza, że wyrzucono połowę „starego” CPC i zastąpiono ją przerosniętym skalakiem 40489, który dodatkowo steruje rozszerzonym dźwiękiem i grafiką, oraz zastosowano uniwersalną płytę główną, wspólną dla 464 i 6128 (w przypadku 464 Plus, po prostu nie montuje się niektórych układów). W dziurę wtykamy magazynek, zawierający w środku 128K ROM oraz 16-to nóżkowy skalak produkowany tylko przez firmę Amstrad, bez którego nie da się wyprodukować ma-

gazynka — w ten sposób firma może kontrolować produkcję oprogramowania. Gra z magazynka ma do dyspozycji niby ten sam dźwięk, ale obsługiwany przez DMA (bez przemęczania starego Z80), rozdzielczość podobna do Atari ST (brak dokładnych danych), 4096 kolorów (max. 16 w jednej linii), 16 sprajtów (16 na 16 pikseli, 16 kolorów innych niż dla tła), płynny scrolling sprzętowy o 1 piksel w dowolnym kierunku, joystick analogowy oraz zwykły. Niestety, wszystkie te super extra możliwości są dostępne TYLKO z magazynka, co nie wróży wielkiej przyszłości „Plusom”.

Komputery serii PLUS są zgodne programowo z serią CPC, pewne niezgodności występują na poziomie sprzętowym (zmieniona kolejność wyprowadzeń złącza dodatkowej stacji dysków). Można wykorzystać wszystkie stare programy i większość urządzeń peryferyjnych. Drukarka ma w końcu 8 bitów, a wszystkie złącza (szyna systemowa, port drukarki, złącze dla drugiej stacji dysków) to gniazda CANON.

Ceny: konsola GX4000 kosztuje ok. 100 funtów, 6128 Plus około 350 funtów (monitor mono). Do każdego egzemplarza dodawany jest magazynek z grą „Burnin Rubber”. Cena magazynka z programem nie jest znana, ale wiele firm planuje wypuszczenie rozbudowanych wersji gier.

(MS)

# WNICA

kroprocesor steruje nie tylko pracą frankownicy, ale także liczy ostemplowane listy.

Co najciekawsze użytkownik ma możliwość umieszczenia poza znacznikiem na kopercie np. własnego znaku lub hasła reklamowego. Jeśli redakcja „Bajtki” wejdzie w posiadanie frankownicy, to wtedy nasze listy opatrzymy gustownym napisem — „Bajtek to Twój najlepszy przyjaciel”.

M.C.



## SPELL MASTER

**Widoczny** na zdjęciu komputer SpellMaster firmy Franklin Computer może okazać się pomocny w sytuacji, gdy twój angielski nie jest jeszcze poparty egzaminem państwowym.

Wystarczy wyklepać na ciekłokrystalicznym wyświetlaczu wymowę wątpliwego słowa, by za chwilę ujrzeć jego pisownię, np. Masheen — Machine. Pozwoli to uniknąć błędów, szczególnie rażących w ważnych dokumentach.

Oprócz tego, SpellMaster posiada kilka możliwości, które oparte na zapisaniu w jego pamięci 90000 pozycyjnym słowniku Merriam-Webster oraz szybkim przetwarzaniu rozwiążą wiele codziennych kłopotów.

Możliwe jest zapoznanie się z wymową poszczególnych słów, poszukiwanie wyrazów bliskoznacznych, dopasowywanie do krzyżówek słów na podstawie kilku liter, specjalna edycja wyrazów łączonych.

Użytkownik może zapisać w pamięci 50 osobistych słów, nawet spoza istniejącego słownika.

A w chwilach nudy można zagrać w jedną z piętnastu gier słownych — anagramy, zgadywanki, test na pamięć i powtarzalność i wiele innych.

(MP)



## STACJA DYSKÓW GIGAFILE

**Stacja dysków** CSS-Gigafile 650 jest jednym z ciekawszych rozwiązań technicznych, które zastosowano ostatnio w dziedzinie pamięci masowych. Nośnik, na którym zapisuje się informacje, bardzo przypomina wyglądem znane wszystkim dyskietki elastyczne 3,5". W twardej, plastikowej kasecie nie znajdziemy jednak zwykłego krążka pokrytego warstwą magnetyczną. Mieści się w niej dysk magneto-optyczny 5,25", który po dwustronnym sformatowaniu pozwala zapisać aż 650 MB danych.

Szybkość transmisji danych wynosi dla stacji Gigafile 680 kB/s. Przeciętny czas dostępu do informacji 90 ms, lecz dla 2 MB danych zapisanych na 128 ścieżkach jest on mniejszy niż 20 ms. Kasecja z dyskiem magneto-optycznym jest odporna na działanie pól magnetycznych i wstrząsy.

Zapisany dysk można w każdej chwili wymienić na nowy lub schować do szuflady. Ma to niebagatelne znaczenie ze względu na ochronę danych. Stacja dysków w pełni odpowiada normom ISO.

Cena Gigafile 650 jest niemała: 9980 DM za stację dysków, 398 DM zestaw kabli i oprogramowania umożliwiającego przyłączenie do ST i 648 DM za dodatkowe dyski. Może ona początkowo odstraszać nabywców. Ale czy to jest rzeczywiście dużo? Za twardego dysku o pojemności 210 MB dealerzy w RFN żądają przecież od 2.400 do 3.000 DM, a za 700 MB około 6.000 DM. Łatwo więc obliczyć, że proporcje cenowe przy korzystaniu z kilku wymienionych dysków okazują się bardzo korzystne.

(JJ)

## DYSK TWARDY DO PCW

**Nowością** angielskiego rynku 8-bitowych komputerów Amstrad są dyski twarde firmy Cirtech o pojemności 20 lub 40 MB. W odróżnieniu od poprzednich konstrukcji charakteryzują się one tym, że są montowane bezpośrednio do komputera w pobliżu złą-

cza szyny systemowej. Zastosowanie bardzo oszczędnych energetycznie napędów 2,5" pozwoliło na istotne zmniejszenie rozmiarów całego urządzenia, którego wygląd zewnętrzny przypomina opisywane poprzednio w Bajtku interface'y do Amstrada PCW. Zniechęca trochę wysoka cena tego gadżetu — 499 funtów (20MB) i 599 funtów (40MB).

(JM)

## MAKRO

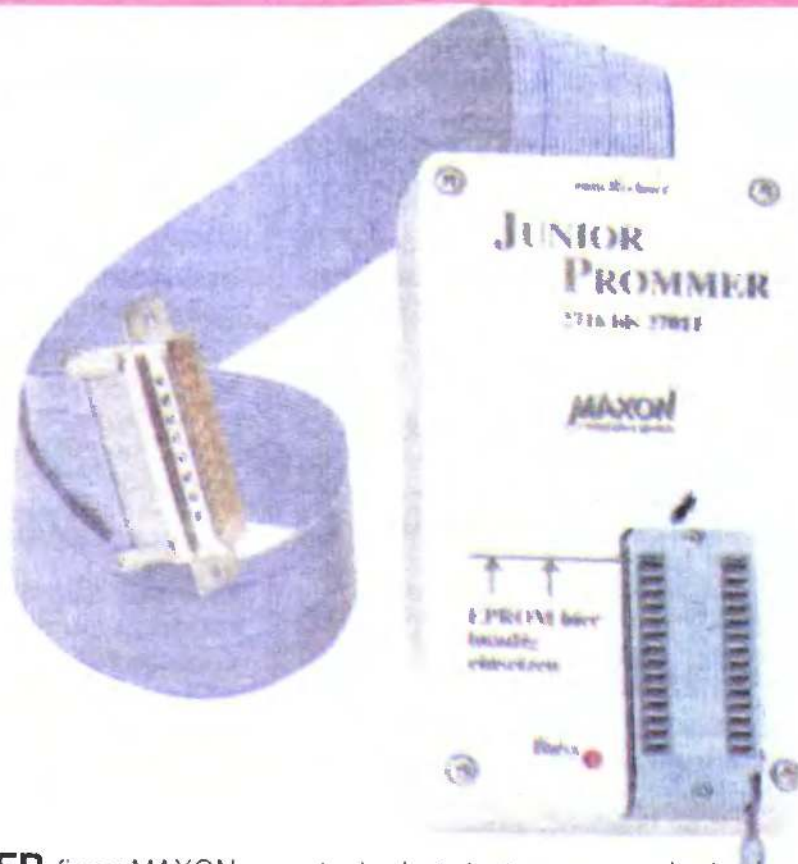
**Chociaż** kolumna nazywa się mikro-ciekawostki, tym razem będzie w niej coś makro i to z dwóch powodów. Po pierwsze, informacja dotyczy dużego, „prawdziwego” komputera, nie mającego nic wspólnego ze sprzętem domowym, po drugie, mowa będzie o dużych pieniądzach. Jeden z brytyjskich banków (którego nazwy nie ujawniono), przekazał przypadkowo w ciągu pół godziny na konta obsługiwanych przez siebie amerykańskich i angielskich firm dwa biliony funtów, czyli ponad trzy biliony dolarów. Wszystko dlatego, że program obsługujący

przepływał pieniądze między kontami pozwalał (w specyficznych okolicznościach) na powtórzenie niektórych przelewów. Większość pieniędzy udało się odzyskać dzięki dobrej woli obdarowanych, ale niektórzy uznali, że pieniądze spadające z nieba na ich konta należą do nich. Nic nie wskazuje na to, że uda się od nich odzyskać pieniądze — po pierwsze, nie pozwalają na to przepisy i ustalenia dotyczące sieci bankowej, w myśl których dokonanie przelewu jest operacją nieodwracalną, po drugie — nie udało się odtworzyć pełnej listy kont, na które dokonano podwójnych wpłat.

Jak narazie w Polsce mamy problemy zupełnie innego rodzaju — u nas płacą nie za dużo, tylko za mało, i nie w ciągu pół godziny, a w ciągu pół roku.

(mb)

## PROGRAMATOR EPROM



**JUNIOR PROMMER** firmy MAXON Computer GmbH przyda się szczególnie tym, którzy mają smykałkę do majsterkowania. Wydając stosunkowo niewielką sumę pieniędzy zyskujemy możliwość samodzielnego oprogramowania układów EPROM, które są niezbędne w najróżniejszych układach elektronicznych.

Programator Junior pozwala zaprogramować wszystkie rodzaje typowych EPROM-ów, począwszy od 2716 (2 kB), a kończąc na nowoczesnych 27011 (1MB). Postępując się Juniorem można także odczytać zawartość niektórych kości ROM i zaprogramować pewne rodzaje układów EEPROM. Programator wymaga zasilania +5V. Wszystkie inne napięcia wymagane podczas wypalania EPROM wytwarzane są przez jego elektroniczny układ. Za wykrywanie błędów i poprawność zapisu w każdym EPROM-ie odpowiada pięć algorytmów kontrolujących dane. Przeglądanie zawartości układu (w postaci heksadecymalnej lub ASCII) i wprowadzanie poprawek umożliwia program monitor.

Junior jest dostarczany w obudowie wraz z odpowiednim kablem. Może on współpracować z Atari ST, Amigą i komputerami z systemem operacyjnym MS-DOS. W przypadku ST urządzenie przyłącza się do portu joysticka. W skład wyposażenia standardowego wchodzi również dyskietka z oprogramowaniem. Znajdziemy tu program zarządzający ramdyskiem, podprogramy odczytu i zapisu danych oraz program ułatwiający konstruowanie kart z pamięciami EPROM. W skład wyposażenia standardowego wchodzi także instrukcja obsługi i schematy elektryczne.

Obowiązująca w RFN cena urządzenia w wersji dla ST wynosi 229 DM, dla Amigi 249 DM, a dla IBM-a 348 DM. Nie jest więc ona wysoka w porównaniu z cenami różnego rodzaju kart rozszerzających możliwości komputera i sterowników, których na ogół nie można samodzielnie zbudować bez programatora.

(JJ)

# PROGRAMOWAĆ MOŻE KAŻDY JĘZYK



DLA  
NAJMŁODSZYCH

czyli dziewiąty wieczór z czarnoksiężnikiem

Dziś czarnoksiężnik przypomniał sobie, że podczas trzeciego wieczoru zaistniała konieczność stworzenia podręcznego notatnika. Niestety, wtedy umiał niewiele i pracę tę odłożył na później. Dopiero teraz może przystąpić do jego realizacji.

Przede wszystkim, muszę ustalić, jakie informacje mają stanowić treść notatnika. Oczywiście jest, że będę przechowywał dane dotyczące moich znajomych. Będą to: nazwisko, imię, adres, telefon, dzień imienin. Chciałoby się, żeby wszystkie dane dotyczące jednej osoby stanowiły całość.

Czarnoksiężnik wziął księgę i zaczął czytać: „Język «C» ma możliwość grupowania kilku wartości w jedną strukturę. Mogą one być różnych typów. Deklaracja struktury, w jaką się one układają, jest jakby definicją nowego typu:

```
struct
{tekst  nazwisko;
 tekst  imie;
 tekst  telefon;
 calkowita  dimien [2];
};
```

Za nawiasem klamrowym, zamykającym opis struktury, następuje lista zmiennych, będących taką strukturą.

Do definicji jakiejś struktury możemy odwoływać się w kilku miejscach programu. Warto wtedy jej opis jakoś nazwać i później powoływać się na tę nazwę, np.

```
struct osoba
{tekst  nazwisko;
 tekst  imie;
 tekst  telefon;
 calkowita  dimien [2];
};
```

i później deklarować: **struct osoba x;** Oczywiście, po „nazwanej” deklaracji struktury może także wystąpić lista zmiennych.

Sama deklaracja struktury nie powoduje rezerwacji pamięci dla tej struktury. Jest tylko informacją dla kompilatora o „architekturze” struktury. Dopiero „przywiązanie” do struktury jakiejś listy zmiennych zarezerwuje pamięć dla zmiennych z tej listy.

Do każdej zmiennej, będącej częścią składową struktury, można mieć dostęp

dzięki operatorowi „kropki”. Dla przykładu, instrukcje:

```
x.dimien[0] = 13;
x.dimien[1] = 5;
```

spowodują, że w naszym notatniku, osoba x ma imieniny 13 maja, a

```
x.nazwisko = „Kowalski”;
```

określi jej nazwisko jako Kowalski

Polem struktury (częścią składową) może być zmienna lub tablica zmiennych dowolnego typu — w szczególności, może to być struktura. Gdybyśmy chcieli zapamiętać dzień imienin nie jako dwie liczby (nr dnia i nr miesiąca), lecz jako numer dnia i nazwę (skrót) miesiąca, to możemy to osiągnąć poprzez taką deklarację:

```
struct data
{calkowita  dn;
 tekst      *mc;
};
struct osoba
{tekst  *nazwisko;
 tekst  *imie;
 tekst  *adres;
 tekst  *telefon;
 struct data  dimien;
};
```

W przypadku tak skonstruowanej struktury zapamiętamy datę 13 maja jako dzień imienin osoby x, jeśli wykonamy instrukcje:

```
x.dimien.dn = 13;
x.dimien.mc = „maj”;
```

Ze zmiennych strukturalnych, tak jak ze wszystkich innych zmiennych, można tworzyć tablice. Zapis **struct osoba x[5];** jest deklaracją pięcioelementowej tablicy struktur typu „osoba”.

Podobnie jak dla zmiennych „normalnych”, operator & podaje adres struktury. Deklaracja:

```
struct osoba *p;
```

informuje nas, że p jest wskaźnikiem do struktury typu osoba. Wartością zmiennej p jest więc adres wspomnianej struktury.

Ci czarodzieje, którzy parają się czarami dłuższy czas, zapewne spotkali się z pojęciem listy lub drzewa binarnego. Lista jest to sposób zorganizowania zmiennych tak, aby oprócz wartości zmiennej znany był adres następnego (lub poprzedniego) elementu. Drzewo binarne oprócz wartości zmiennej zawiera adres elementu lewego i prawego (rys. 1).

```
struct  tytul
{tekst  *t;
 struct tytul  *nast;
 struct tytul  *poprzedni;
};
```

Deklaracja:

definiuje nam dwukierunkową listę tytułów książek. Mając taką strukturę, łatwo te tytuły poustawiać w kolejności alfabetycznej. Wystarczy tylko odpowiednio „zonglować” wartościami wskaźników (adresami). Wprawdzie struktura taka zajmuje więcej miejsca w pamięci niż sama zmienna (tytuł), ale zysk na czasie przy przedstawianiu dwu zmiennych jest olbrzymi. Zamiast przepisywać całe tytuły w nowe miejsca, wystarczy tylko przestawić wartości wskaźników do tych tytułów. Rys. 2 pokazuje czteroelementową listę przed zmianą (a) i po zmianie (b) dwu środkowych elementów. Lokalizacja wszystkich czterech elementów nie uległa zmianie — poprzestawiano tylko wartości niektórych wskaźników (rys. 2a i 2b).

Analogicznie wygląda sprawa z dołączeniem nowego elementu do listy. Można to zrobić z zachowaniem dotychczasowego porządku. Wystarczy utworzyć dynamicznie nową zmienną strukturalną typu „tytuł” i wstawić w odpowiednie miejsca stosowne wartości wskaźników. Zmienne dynamicznie obsługują dwie standardowe funkcje. Funkcja **calloc(n, sizeof(obiekt))**; zwraca wskaźnik (adres) do wolnego, aktualnie przydzielonego, obszaru pamięci. Może on pomieścić n obiektów. Rozmiar takiego obiektu podawany jest przez funkcję **sizeof(obiekt)**. Oczywiście, zasoby pamięci systemu operacyjnego są zmniejszane o przydzieloną pamięć. Gdy system nie dysponuje już wystarczająco dużym obszarem pamięci i nie może zadośćuczynić naszym prośbom, wówczas funkcja **calloc** zwraca wartość NULL (zazwyczaj jest to zero binarne).

Funkcja **cfree(p)**; zwalnia pamięć wskazaną przez wskaźnik p, wcześniej dynamicznie przydzieloną przez **calloc**. Możemy w ten sposób unicestwić niepotrzebne zmienne dynamiczne. Oczywiście zwalniana pamięć powiększa zasoby systemu operacyjnego.

Należy jeszcze zastanowić się, jak zrealizować dostęp do poszczególnych pól struktury w przypadku, gdy odwołujemy się do zmiennej strukturalnej poprzez wskaźnik do niej. Oczywiście konstrukcja to:

```
(*p).telefon
```

Ponieważ wskaźniki są często używane, wprowadzono operator -> (minus i >). Wykorzystując go, można osiągnąć dostęp do pola struktury wywoływanej przez wskaźnik, dzięki zapisowi:

```
p->telefon
```

operatory strukturalne -> i kropka oraz nawiasy () dla listy argumentów i [] dla indeksów mają najwyższy priorytet. Dlatego instrukcja **++p->x**; zwiększy x, a nie p, a **\*p->x** jest wskaźnikiem do obiektu wskazanego przez x, a więc **(\*p->x)++**; zwiększy obiekt wskazany przez x.

Priorytet operatorów został przedstawiony w tabeli.

## Tabela

operatory wg priorytetu: najniższy	łączność
najwyższy	L — lewostronna P — prawostronna
() [] .->	L
! ~ ++ -- (typ) * & sizeof	P
* / %	L
+ -	L
< >	L
< <= > >=	L
== !=	L
&	L
^	L
	L
&&	L
	L
?:	P
= += -= itp	P
,	L

**Operator (typ) zmienia typ argumentu na deklarowany w nawiasie np. (rzeczywista) 5 zmienia typ liczby całkowitej 5 na liczbę rzeczywistą 5.0. Pociąga to za sobą inny sposób zapamiętania tej liczby.**

Wszystkie pola struktury mają swoje oddzielne miejsca w pamięci i wszystkie jednocześnie istnieją. Każda zmienna, w szczególności pole struktury, jest pewnego typu i tylko tego typu wartość może przechowywać. Zdarza się jednak, że ta sama zmienna, w różnych sytuacjach, musi mieć inny typ i rozmiar. Wartość każdego typu musi być umieszczona w tym samym miejscu i dostępna jako ta sama zmienna. Każdy typ zajmuje inną ilość komórek pamięci i ważne jest, żeby zarezerwowane miejsce pomieściło każdy zadeklarowany typ. W tej sytuacji wszystkie typy układamy w strukturę. Żeby jednak każdy typ był lokowany w tym samym obszarze pamięci, struktura ta tworzy unią. Zacytujmy przykład deklaracji unii:

```
union tag
{calkowita  ival;
 rzeczywista  rval;
 tekst  pval;
 uval;
```

Unia jest zmienną, która w różnych momentach może przechowywać obiekty różnych, wcześniej zadeklarowanych, typów (więc i rozmiarów). Zmienna będąca unią jest takiego typu, jaki został ostatnio do niej wstawiony. Kontrola typu zmiennej, której wartość aktualnie przechowuje unia, należy do programisty.

Wszystkie prawa obowiązujące dla struktur dotyczą także unii. Można więc mówić o wskaźniku do unii, a także o tablicach unii. Unia może być, w wyniku zagnieżdżenia, polem jakiejś struktury.

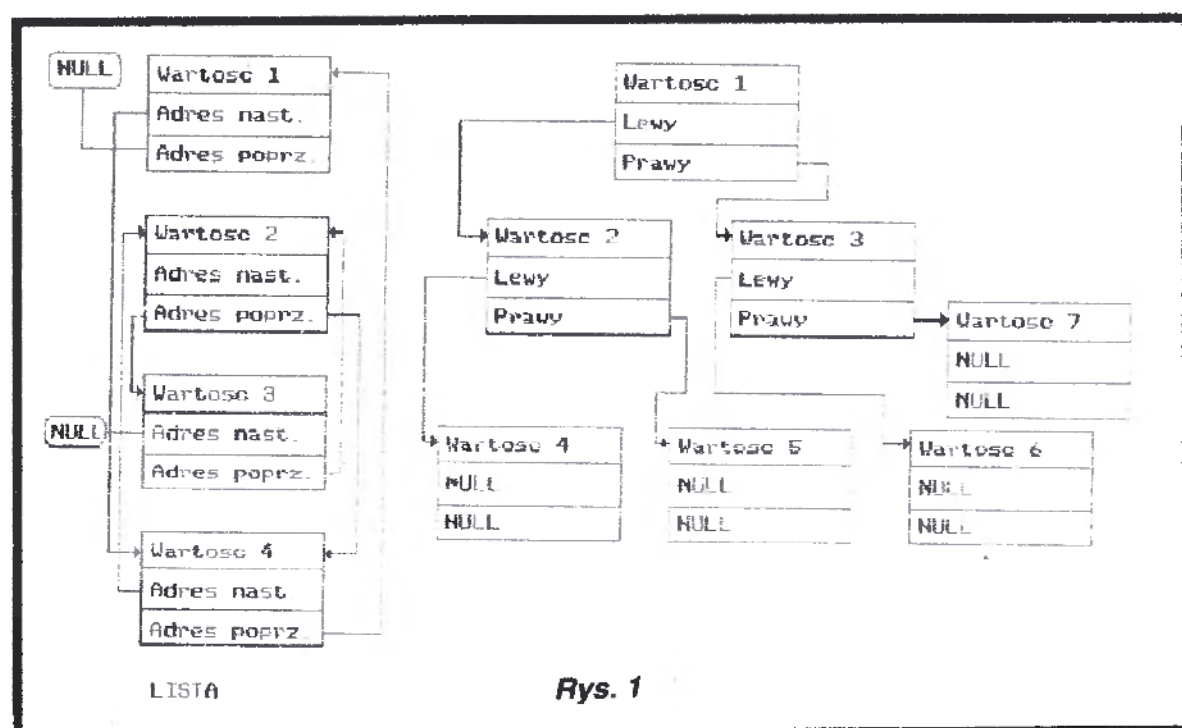
Wywołanie unii z jakimś typem można osiągnąć dzięki znanym już operatorom kropki i ->.

```
nazwa_unii.skladowa
wskaznik_unii->skladowa
```

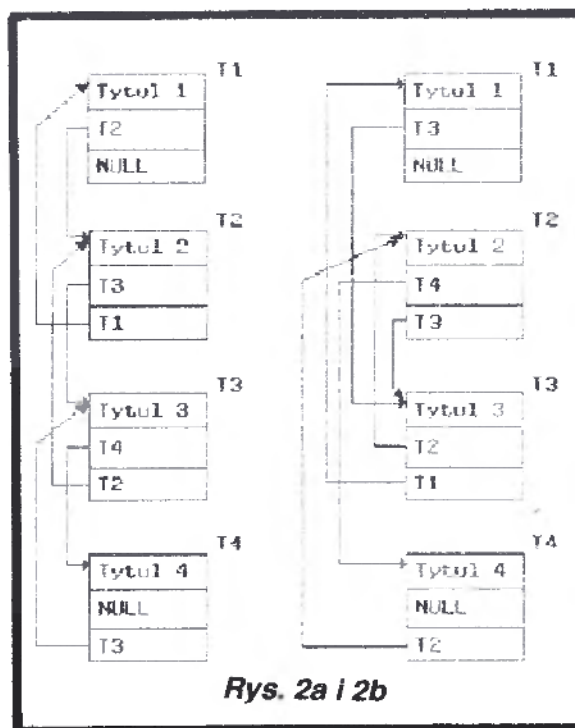
Błędem będzie, jeśli typ wywoływanej unii (składowej) będzie niezgodny z typem obiektu ostatnio w niej umieszczonego.

Niestety, dzisiejsza nauka była sztuką dla sztuki — pomyślał czarnoksiężnik. Bardziej dorosłe kompilatory języka „C” akceptują wszystkie konstrukcje omawiane w naszym cyklu. Ja dysponuję kompilatorem „Deep Blue C” na małe Atari, który akceptuje tylko okrojona wersja tego języka. Żeby mieć możliwość pracy z każdym innym kompilatorem, poznaliśmy wszystkie jego reguły. Reszta zależy od konkretnej implementacji.

Mieczysław Płacheta



Rys. 1



Rys. 2a i 2b

# PO CO BELFROWI KOMPUTER?

**Nikom, a zwłaszcza nikomu z czytelników „Bajtki”, nie trzeba tłumaczyć, jakim dobrodziejstwem są komputery — czego to dzięki nim nie można dokonać i jak trudne jest życie bez nich. A gdy wreszcie nadejdzie dzień, kiedy zasiądziemy przed własną klawiaturą, przeważnie odczuwamy rozczarowanie. Bo po pierwsze, nie mamy odpowiednich programów, a bez programów komputer jest niczym, a po drugie — najczęściej niedokładnie wiemy, czego od komputera oczekujemy.**

Brak programów jest problemem, którego można zaradzić. Krótkie, łatwe programy (a raczej programiki) możemy pisać sami, po niewielkim trudzie włożonym w naukę podstaw programowania. Programy dłuższe i dużo lepsze możemy kupić (lub raczej powinniśmy móc kupić) w firmach produkujących oprogramowanie.

Jednak, aby móc wybierać, musimy wiedzieć, jakich programów potrzebujemy, czego możemy wymagać, a co leży poza możliwościami komputera (lub naszej kieszeni).

Przed tymi samymi problemami staje też nauczyciel-użytkownik komputera, bo (choć może niektórzy uczniowie się zdziwią) nauczyciel to też człowiek, i to człowiek, który nie ma wiele czasu na naukę innych przedmiotów niż ten, który z wielkim nakładem pracy wykłada. Dlatego w niniejszym artykule postaramy się przedstawić, czego może oczekiwać zwykły belfer, który chcąc uczyć „zgodnie ze swoją najlepszą wiedzą” wprowadza do procesu nauczania komputera.

Na lekcji można komputer używać w dowolny sposób: jako sprzęt do ćwiczeń dla uczniów albo jako pomoc wizualną dla nauczyciela, taką samą jak tablica, plansza, rzutnik pisma lub przezroczny czy projektor. Zajmiemy się tą drugą możliwością, bo pierwsza wymaga większej ilości sprzętu, a na to stać niewiele szkół. Wykorzystanie komputera jako pomocy wizualnej wymaga tylko telewizora (najlepiej dużego), niemal dowolnego komputera (na początek dla mniej zasobnych szkół wystarczy SPECTRUM, ATARI, AMSTRAD lub inny popularny komputer domowy). Jeśli szkoła nie ma sprzętu, możemy poprosić uczniów, na pewno którzyś z nich ma komputer i tylko marzy, by się nim pochwalić.

Jak wynika z doświadczeń autora, sprzęt musi być wyposażony w stację dysków. Wgrywanie programów z taśmy magnetofonowej trwa bardzo długo (i musi tyle trwać), a co znaczy na lekcji każda zaoszczędzona minuta nie muszą chyba mówić żadnemu nauczycielowi-praktykowi. Jeśli nie mamy i nigdzie nie możemy zdobyć stacji dysków, dobrze jest wykorzystywać program wgrać przed lekcją (choć szkoda przerwy) i wyłączyć telewizor do czasu użycia programu. Inne sposoby pracy powodują zbyt dużą w stosunku do zysków dezorganizację lekcji.

Tyle o sprzęcie. A co może być w sprzęcie, czyli jak go wykorzystywać?

Każda z pomocy naukowych ma swoje zalety i wady. Np. zaletą pospolitej tablicy jest możliwość szybkiego tworzenia napisów i odręcznych szkiców, a wadą, że szkice te właśnie są odręczne, a więc niedokładne i nieprecyzyjne. Plansze z kolei są precyzyjne i dokładne, ale za to nic z nimi nie można zrobić (prócz pokazania oczywiście). Dobrze napisany program komputerowy nie ma prawie żadnych wad, gdyż robi właśnie to, co chcemy, aby robił.

Prześledźmy możliwości wykorzystania komputera na przykładach. Nie chodzi tu o gotowe programy, lecz o możliwe scenariusze programów. Proponowane lekcje dotyczą:

— z historii: kampania 1409—1410 i bitwa pod Grunwaldem,

— z geografii: epoki lodowcowe i nachełdzenie lodowców,  
— z fizyki: prawo Archimedesesa,  
— z matematyki: funkcje trygonometryczne SINUS i COSINUS,  
— z chemii: tablica Mendelejewa.

Zastosowanie komputera jako tablicy i plansz polega na tym, że na ekranie pojawiają się statyczne rysunki obrazujące wykład i uzupełniające nasze słowa. Zysk polega na tym, że możemy nie tylko wskazywać elementy składowe, lecz również podświetlać je, zaznaczać i akcentować w inny sposób. I tak:

— nauczyciel historii ma do swojej dyspozycji komplet map państw uczestniczących w konflikcie, jak i ważniejszych regionów (przeprawy przez Wisłę i Drwęce, Pole Grunwaldzkie itp.), przedstawiane w dowolnej formie (tabele, wykresy) siły wojsk, uzbrojenie, nazwiska dowódców itp. wraz z rysunkami wojów. Może na mapę narzucić położenia wojsk dowolnego dnia, zaznaczyć trasy przemarszu itp.;

— nauczyciel geografii ma również do dyspozycji komplet map ziemi (oczywiście z interesującego go okresu) z zaznaczonym położeniem lodowców w wybranym okresie. Na mapy te może nakładać inne potrzebne mu informacje (średnie izotermie, ciśnienie wywierane przez lodowiec, stężenie CO<sub>2</sub> w atmosferze itp.), a nadto jak w formie tablic lub wykresów może przedstawiać takie elementy, jak np. średnie temperatury w danym okresie, czas, szybkość i kierunki ekspansji lodowca, jego łączną powierzchnię itp.;

— nauczyciel fizyki dysponuje rysunkiem ciała zanurzonego w cieczy wraz z zaznaczonym rozkładem sił, możliwością pokazania rozkładu ciśnień, wartościami gęstości ciał i cieczy oraz położenia ciała przy zadanych wartościach gęstości.

— nauczyciel matematyki przedstawić może wykresy żądanych funkcji rysowane pojedynczo bądź nakładane na siebie. Może również pokazać bez trudności wykresy dowolnych kombinacji tych funkcji (suma, różnica bądź iloraz funkcji, suma kwadratów, wykresy dowolnej potęgi, mnożenie przez stałą funkcji i jej argumentu itp.);

— nauczyciel chemii wreszcie ma do dyspozycji nie tylko rysunek całej tablicy Mendelejewa, ale może wyróżniać w niej określone grupy pierwiastków, prezentować wszystkie dane o pierwiastkach, a także wyznaczać zbiory pierwiastków spełniające określone warunki (np. po poleceniu wyszukania atomów o określonej liczbie elektronów na powłoce walencyjnej okaże się, że tworzą one jedną grupę, co znakomicie ułatwi prezentację zasad rządzących tablicą).

Prawdziwa moc i siła dydaktyczna programów objawi się dopiero po dodaniu do rysunków elementu czasu — ożywienie plansz i rysunków na tablicy. Dopiero ten zabieg przeniesie nauczycieli w zaczerpnięty świat, w którym uczniowie są zasłuchani w ich słowa i można im przekazać wszystkie niuanse poruszanych zagadnień. Tak powstaje możliwość wykorzystania komputera jako udoskonalonego projektora filmowego. Udoskonalonego, gdyż użytkownik jest nie

tylko obserwatorem, ale również współtwórcą oglądanego filmu.

Wyobraźmy sobie nauczyciela historii, który na omawianej lekcji ma możliwość przedstawienia na ekranie jednocześnie ruchów wszystkich wojsk, ich położenia danego dnia wraz z możliwością ciągłego „monitoringu” zapasów żywności, siły koni i co za tym idzie konieczności robienia popasów, czasu potrzebnego na skontaktowanie się z siłami sojuszniczymi. Jak ciekawa będzie dla uczniów obserwacja sytuacji na polu bitwy i przebieg walki w dowolnej chwili wraz z możliwością unaocznienia siły i możliwości bojowych oddziałów, przesyłaniem rozkazów przez gońca itp. Możliwe jest nawet (choć jest to trudne dla programisty — ale za to mu właśnie płacą) przeprowadzenie „alternatywnej” kampanii i bitwy przez uczniów i nauczyciela. Możemy sami stanąć na miejscu Jagiełły bądź Wielkiego Mistrza i wydawać rozkazy tak podczas marszu jak i samej walki. Program, biorąc pod uwagę wszystkie uwarunkowania, przedstawi sytuację po wydaniu tej czy innej komendy. Okazuje się wówczas nie tylko, że tę wojnę mogliśmy przegrać, jeśli nie dorównujemy geniuszowi Jagiełły (a może doprowadzilibyśmy do zwycięstwa mniejszym kosztem?). Okazuje się także, że historia jest ciekawa i tworzą ją ludzie, a nie jest tylko zbiorem nudnych faktów i dat, jak wyobraża sobie większość uczniów.

Nauczyciel geografii może nie tylko sterować biegiem czasu i pokazywać rozwój sytuacji w zamierzonych epokach. Może także jednocześnie pokazywać rozwój w czasie innych wybranych wielkości, jak np. zasięg występowania życia. Uwidocznic to może, że żyjemy po prostu w kolejnym interglacjale i być może grozi nam kolejna epoka lodowcowa. Można — w odpowiednio skonstruowanym programie — zmienić warunki panujące w dowolnym okresie, powiedzmy — wprowadzić w pewnej chwili więcej dwutlenku węgla do atmosfery, co zwiększy efekt cieplarniany i temperaturę, i obserwować rozwój sytuacji. Można na drodze lodowca „ustawić” pasmo górskie i zobaczyć, do jakich zmian to doprowadzi. Można też przejść na skalę „mikro” i sprawdzić, jakie siły wywiera na podłożu czoło lodowca oraz jak tworzy się morena czołowa i denna przy różnym ukształtowaniu warstw geologicznych.

Fizyk może nie tylko zmieniać gęstość cieczy i ciała w niej zanurzonego. Może też tworzyć warstwy cieczy o różnej gęstości, zmieniać siłę grawitacji i w każdym przypadku obserwować zachowanie się układu, rozkład sił, ich wartości i kierunki, przedstawić rozkład i gradient ciśnień w cieczy, siły wywierane na różne części ciała zanurzonego i inne elementy, o których często nawet nie wspomina się w czasie typowej lekcji.

Dla nauczyciela matematyki możliwość wprowadzenia ruchu ma może mniejsze znaczenie (prawa matematyki są zbyt doskonałe, by zmieniać się w czasie). Jednak w przypadku danego tematu warto być może byłoby przedstawić tworzenie wykresu przy pomocy trójkąta prostokątnego wpisanego w okrąg, który to sposób stosuje się teraz, co jednak w wersji statycznej nie jest zbyt czytelne ani przemawiające do wyobraźni.

Chemik mógłby zainteresować uczniów nie tylko przedstawianiem danych. Można, w odpowiednio skonstruowanym programie, przedstawić rysunki atomów z zaznaczonymi orbitalami i „uwspólnianie” orbit elektronów przy łączeniu się w związki chemiczne. Tworzone tak nowe związki chemiczne, podobieństwa i różnice w „wyglą-

dzie” cząsteczek mogą unaocznic, dłaczęgo pierwiastki z jednej grupy mają podobne właściwości, jakie pierwiastki i jak ulegają rozpadowi, dłaczęgo izotopy nie różnią się od siebie pod względem chemicznym, a pierwiastki szlachetne niechętnie łączą się w związki.

Jak widać choćby na przykładzie tych niekompletnych propozycji, od komputera może nauczyciel wymagać ułatwień w trzech dziedzinach.

Po pierwsze, jest to możliwość uzyskania wszelkich rodzajów danych w różnych postaciach. Pod pojęciem danych rozumiemy nie tylko wartości stałych, liczb i wartości tabelaryzowanych. Dane komputerowe obejmują również rysunki, fazogramy, teksty, mapy, a także zdjęcia prezentowane na ekranie, filmy animowane przedstawiające pracę urządzenia lub przebieg zjawiska, a prócz tego wszystkie dane, jakie możemy sobie wyobrazić.

Po drugie, od programów można wymagać możliwości ingerencji w prezentowane obrazy. Możliwe jest stworzenie programu, który pokaże rozwój struktury i sytuacji po podaniu przez nas tylko warunków początkowych. Jest to łatwiejsze w przypadku przedmiotów ścisłych (fizyka, chemia), a znacznie trudniejsze bądź nawet niemożliwe np. w historii, gdzie zbyt dużo uwarunkowań, z których część będąc nieokreślona, ma wpływ na sytuację. Można napisać program pokazujący przebieg zjawiska bądź w postaci filmu, bądź przez demonstrację wskazań przyrządów. Możemy wręcz stworzyć układ demonstracyjny lub pomiarowy z dostępnego w komputerze zestawu elementów. Trzeba jednak uprzedzić, że program taki jest trudny, a więc i drogi.

Po trzecie, możemy żądać, by program miał możliwość przedstawiania sytuacji przy zmienionych warunkach lub prawach. Przy odpowiedniej konstrukcji programu możliwe będzie wręcz tworzenie nowych światów wraz z prawami w nich obowiązującymi. Lekcja z takim programem (typ ten jest zwany TEACHING ASSISTANT MODE) wygląda tak, że nauczyciel po przedstawieniu problemu wysłuchuje propozycji rozwiązania podawanych przez uczniów i wprowadza je do programu. Rozwój akcji potwierdza bądź obala te hipotezy.

Programy mogą być różne. Raz są to wielkie pakiety programowe, świetnie napisane i bardzo uniwersalne. Inne programy są krótkie, przeznaczone do rozwiązania tylko jednego, konkretnego problemu. Wydaje się (autorowi), że właśnie takie krótkie programy, przeznaczone do wykorzystania na jednej konkretnej lekcji, są przydatniejsze dla nauczyciela. Gdzie jednak zdobyć takie programy? Niestety jest na to tylko jeden sposób: trzeba zmusić firmy do ich produkcji. Zmusić przez wytworzenie chłonnego i żywo reagującego rynku zbytu. Kto ma płacić za te programy, szkoły czy MEN, jest to temat na zupełnie inny artykuł i do tego w zupełnie innym piśmie. A nie łudźmy się: płacić trzeba, gdyż inaczej komputery będą wyjmowane z dna szkolnych szaf w czasie wizytacji kuratorium.

Nie wspomnieliśmy w tym artykule o innych możliwościach wykorzystania komputerów: do testów, jako podręczniki programowane, dzienniki, zbiory zadań, do indywidualnej pracy uczniów i na wiele innych sposobów. Postaramy się przedstawić wszystkie te problemy w przygotowywanym w „Bajtku” cyklu „PO DZWONKU”, prezentującym problemy dydaktyki.

T.B. Mańk

# AST

## drugie spojrzenie

**Dotychczas o Ast napisano wiele ciepłych słów, czas więc na nieco krytyki. Ten, w całości opracowany w Polsce, system służący do szybkiej transmisji danych z magnetofonu posiada wiele oryginalnych i ciekawych rozwiązań, ale równie wiele wad, o których warto wiedzieć.**

Jedną z podstawowych zalet dostarczonego zestawu programów jest sposób transmisji, która odbywa się w postaci jednego bloku — bez przerw. Przyjęty sposób organizacji przesyłania danych, oprócz oczywistych zalet ma kilka poważnych wad. Pierwsza to brak zainstalowanej procedury obsługi urządzenia "T:", co już na starcie prawie wyklucza korzystanie z programów użytkowych (Koala wczytuje się w turbo, ale rysunki zapisuje standardowo — rysunki zajmują kilkadziesiąt razy więcej miejsca niż sam program!). Drugim, ubocznym tego

efektem jest zmiana formatu zapisywanych danych, przestaje on być zgodny ze standardem DOS-u. Jest to o tyle przykre, iż powoduje, w połączeniu z brakiem zainstalowanej procedury obsługi, niemożność korzystania z jakichkolwiek programów narzędziowych, co jest wykonalne w TURBO 2000 czy BLIZZARDzie. Co gorsza, Atari Studio nie zdołało stworzyć jednego formatu zapisu danych — zupełnie inny jest format programów kopiujących, procedury obsługi "T:" i pakietu dodatkowego, co powoduje dodatkowe poważne trudności. Tym niemniej można by mówić, że jest tu coś za coś. Program nie zajmuje obszaru DOS-u (to ogromny plus) i wczytuje się bez przerw, ale za to mamy powyższe trudności. Niestety, oprócz wyrzeczeń wynikających niejako z filozofii systemu mamy również bardzo przyziemne niedoróbki i wręcz radosne błędy, o których poniżej.

### PROGRAMY KOPIUJĄCE

Niewątpliwie największą niespodzianką czekającą użytkownika w trakcie kopiowania programów na standard turbo jest brak wariantu czyszczenia bufora w dostarczonych programach kopiujących. W praktyce oznacza to, że po skopiowaniu czegoś należy wczytać je ponownie. Proszę wyobrazić sobie, że mamy program wczytujący się za przysłowiowym 10 razem. Po każdym przekłamaniu odczytu trzeba wyłączać komputer i ładować program kopiujący! Czas kopiowania wydłuża się z minut do godzin, nie mówiąc o irytacji użytkownika. Za drobny błąd należy uznać niedziałanie klawisza DELETE w trakcie wpisywania nazwy programu. W dodatku po wpisaniu czegoś do komórek \$2E2 i \$2E3 gry uruchamiają się na programie kopiującym. Co prawda, istnieje do nich specjalny program kopiujący, ale pozostaje pytanie, którego programu użyć (proponuję rzut monetą)\*. Żeby było śmieszniej spora grupa programów nie daje się niczym skopiować. Budzi to szczególną radość, gdy po próbie skopiowania jednym.

a proszę nie zapominać, że trochę to trwa, nie udaje się to również przy pomocy drugiego programu.

### AST-BASIC

Jak na firmę AS jest to program napisany wyjątkowo niechlujnie. Radzę po załadowaniu wykonać POKE 9,0, bo inaczej komputer zawiesi się w najmniej spodziewanym momencie. Kolejne „miłe” zaskoczenie, to sposób wykonania rozkazu LIST "T:filename" — transmisja trwa dłużej niż LIST "C:". Dla mnie jednak rewelacją stanowiła transmisja danych przez blok IOCB. Po podaniu komendy CLOSE moim oczom ukazał się zdumiewający komunikat: ERROR 146 (Function not implemented). Okazuje się, że można tę ustarkę obejść wykonując CLOSE dwa razy, za pierwszym razem przechwytyjąc błąd poprzez TRAP. Jest to jednak mały kłopot w porównaniu z próbą odczytu danych. W odróżnieniu od procedury obsługi "C:" należy podać liczbę danych z dokładnością co do bajtu. Pomyłka może oznaczać w optymistycznych warunkach utratę procedury obsługi "T:". Należy więc w praktyce zapisać na kasetę długość bloku, choć przyznam się szczerze, że nie rozumiem dlaczego procedura nie robi tego automatycznie.

### INNE KŁOPOTY

Dostarczona przez AS wersja TURBO BASICA wykazuje dziwne zachowanie po naciśnięciu INKEY\$ i GET KEY, gdyż oczekuje na naciśnięcie klawisza RESET. Niekiedy kilkakrotne naciśnięcie ESC w AST-LOAD może prowadzić do autodestrukcji programu. „Obszerna” instrukcja jest w gruncie rzeczy napisana mętnie i niewiele z niej wynika.

?

Zagadką jest dla mnie pakiet specjalnych programów do kopiowania na AST. Nie

spotkałem nikogo, kto skopiował by tym cokolwiek. W tej sprawie zwróciłem się nawet do AS, ale ich uwagi nie wyjaśniły niczego. Tego punktu nie jestem jednak pewny, dlatego że może popełniłem jakieś błędy w obsłudze, a jak wspominałem instrukcja jest, deliktanie mówiąc, niespecjalna.

### PODSUMOWANIE

Nie piszę tego, by kogokolwiek zniechęcać. AST jest systemem bardzo ciekawym i wygodnym, o unikalnych rozwiązaniach, umożliwiającym kopiowanie programów, które w żadnym innym systemie na turbo się nie skopiują. O zaletach tego rozwiązania można się przekonać w trakcie użytkowania, ale warto znać również wady. Radą na te ostatnie, jeśli już jesteśmy skazani na magnetofon, jest używanie kilku systemów turbo. Umożliwia to choćby opracowany w Krakowie system Universal Turbo. Zastosowano tam zupełnie odmienne rozwiązania sprzętowe umożliwiające odczyt zarówno w AST, TURBO 2000, BLIZZARDzie, jak i TURBO ROMie. Istnieje również moduł ROM (cartridge) z loaderami do trzech pierwszych systemów. Moja delikatna sugestia pod adresem firmy AS brzmi: może warto zastanowić się nad unowocześnieniem rozwiązań sprzętowych i usunięciem przynajmniej niektórych irytujących błędów w dostarczanych programach.

*Jakub Cebula*

P.S. Opisywałem oprogramowanie przygotowane przez Atari Studio do lata 1989, później już nie miałem styczności z systemem.

\* Można to zrobić TRACER-em lub innym monitorem, ale ponieważ nie można spod nich zapisać w turbo, wychodzi na to samo, co przy próbie czytania na oba programy kopiujące.

# PRZERWANIA W ACTION!

**Programy pisane w języku Action! są tak szybkie, że można je wykonywać nawet podczas przerwania. Dodatkowym ułatwieniem jest tu możliwość umieszczenia procedury maszynowej w dowolnym miejscu programu.**

W przykładowym programie, który jest zamieszczony obok, podczas przerwania TIMER2 wywołana jest procedura **Graj()** odtwarzająca w kółko kilka nut pobieranych z tablicy "melodia".

Procedury i funkcje obsługujące przerwanie to:  
— Funkcja **StartVKT()** zwracająca adres, od którego zaczyna się kod wynikowy procedury podanej jako parametr funkcji.

— Procedura **SetTIMER2()** to procedura, która włącza przerwanie. Trzeba ją wywołać, aby procedura "name", podana jako parametr, była wykonywana w wyniku przerwania. Drugi parametr tej procedury to czas pomiędzy kolejnymi wywołaniami procedury "name" (jednostką jest 1/50 sekundy).

— Procedura **Again()**, o której musimy pamiętać przy pisaniu procedury wywoływanej przerwaniem. Ustawia ona zegar TIMCNT2.

— Procedura **ResetTIMER2()** wyłącza przerwanie TIMER2. Powinna być wywołana przed zakończeniem programu. Zmiany programu, jego ponowna kompilacja i uruchamianie przy włączonym przerwaniu, mogą spowodować zawieszenie komputera.

W podobny sposób można wykorzystać inne przerwania, np. VBLKI, dające równie duże możliwości. Trzeba tylko pamiętać, aby procedura wywołwana przez przerwanie nie była zbyt długa, gdyż może zakłócić pracę systemu.

*Andrzej Postrzednik*

```

;PRZERWANIA W ACTION!
;Andrzej Postrzednik
;(c) 1990, Bajtek

MODULE

CARD TIMECOUNT2=538,rptime

CARD FUNC StartVKT(BYTE ARRAY name)
; funkcja zwraca adres początku
; procedury 'name' po kompilacji

BYTE find,j,size
CARD i,addr
  find=0 size=name(0)
  FOR i=20000 TO 23000 DO
    j=1
    WHILE Peek(i+j)=name(j) DO
      IF j=size THEN find=1 FI
      j==+1
    OD
    IF find=1 THEN EXIT FI
  OD
  IF find=0 THEN
    Print("Nie ma procedury ")
    PrintE(name) Break()
  FI
  addr=PeekC(i+j+1)
RETURN(addr)

; procedura włączająca przerwanie

PROC SetTIMER2(BYTE ARRAY name CARD t)

CARD TIMER2VKT=552
  rptime=t
  TIMER2VKT=StartVKT(name)
  TIMECOUNT2=1
RETURN

PROC Again()
  TIMECOUNT2=rptime
RETURN
; procedura wyłączająca przerwanie

PROC ResetTIMER2()
  TIMECOUNT2=0
RETURN
; koniec procedur
; obsługujących przerwanie

;*****
MODULE

BYTE ARRAY melodia=[96 0 81 0 96 0 96
  72 81 108 96 0 64 96 0 96 60 64
  81 96 81 96 108 128 85 96 0 0 0]
BYTE licznik

PROC Graj()

BYTE glos
  glos=melodia(licznik)
  Sound(0,glos,10,10)
  Sound(1,glos+1,10,10)
  licznik==+1
  IF licznik=29 THEN licznik=0 FI
  Again()
RETURN

PROC Main()
  licznik=0
  SetTIMER2("Graj",10)
  WHILE Peek(53279)#6 DO
    Print("czekam na START ")
  OD
  ResetTIMER2()
RETURN

```



# SORTOWANIU

## JESZCZE O

Do porządkowania tablic liczbowych lub znakowych konieczne jest ułożenie odpowiedniego programu sortującego. Kto próbował to zrobić w Basicu, wie, jak powolny jest taki program. Ze względu na to, że program w języku maszynowym jest prawie sto razy szybszy, najwygodniej jest dołączyć do własnego programu taki program maszynowy, na przykład pokazany tu SORT 2.0.

Program ten jest tak szybki, że 2000 pozycji sortuje w ciągu niecałych 30 sekund, a 200 pozycji w ciągu niecałych 2 sekund. W odróżnieniu od programu "Sortowanie tablicy liczbowej" („Bajtek” 5/78) ma on:

- możliwość sortowania różnych tablic,
- możliwość sortowania tablic znakowych,
- sortuje wszelkie wartości liczbowe,
- jest dziesięciokrotnie szybszy,
- sortuje również tablice dwuwymiarowe, co ma duże znaczenie praktyczne

**SORT 2.0** może także sortować pola liczbowe uwzględniając znaki + i -. Wykorzystuje on znany algorytm sortowania bąbelkowego. Jest to jeden z najszybszych algorytmów.

Właściwy program **SORT 2.0** jest umieszczony w zmiennej tekstowej M\$ przez instrukcje znajdujące się w wierszach 30020-30040. Ponieważ wprowadzanie z klawiatury tych 255 znaków jest bardzo kłopotliwe, został podany również program pomocniczy (listing 2), który należy dopisać do głównego programu. Po jego wprowadzeniu należy podać instrukcję

### GOTO 1037

a komputer sam utworzy wiersze 30020—30040. Po sprawdzeniu poprawności działania całego programu te instrukcje DATA są już niepotrzebne i można je skasować instrukcją

### GOTO 1042

Instrukcja ta kasuje wszystkie wiersze od 1000 do 1042. Uwaga: Zrealizowany w ten sposób program nie będzie działał poprawnie w Turbo Basicu. Pętli, w których wykonywane jest usuwanie wierszy, nie mogą być umieszczone w pętli FOR/NEXT, jak w Atari Basic.

Wiersze od 50 do 220 tworzą program demonstrujący użycie **SORT 2.0** do sortowania jedno- i dwuwymiarowej tablicy liczbowej D (wiersze, kolumny). Dalsza część, od wiersza 230 do 360,

pokazuje sortowanie zbioru znaków umieszczonego w zmiennej tekstowej D\$. Rozdzielenie tej zmiennej na poszczególne pola jest pozostawione dla czytelników.

Procedurę maszynową realizującą sortowanie wywołuje się instrukcją

```
X=USR(ADR(M$), ZS, IP, DL, ZP, KP)
```

gdzie:

ADR (M\$) — adres początku procedury maszynowej umieszczonej w zmiennej M\$;

ZS — adres początku sortowanego zbioru, dla zmiennej tekstowej jest to ADR (D\$); dla tablic liczbowych trzeba przy deklarowaniu użyć instrukcji DIM A\$(1), D (W,K), co powoduje umieszczenie w pamięci tablicy liczbowej bezpośrednio za zmienną A\$, a więc jej adres jest równy ADR(A\$)+1;

IP — liczba sortowanych pozycji (wierszy), UWAGA: tablice liczbowe rozpoczynają się od D(0,0);

DL — dla zmiennej tekstowej jest to określenie podziału zmiennej na pola, a dla tablicy liczbowej — liczba kolumn pomnożona przez 6 (każda liczba zajmuje sześć bajtów);

ZP — pole rozpoczynające sortowanie (liczone od 0), dla pól liczbowych jest to TS\*6, gdzie TS jest liczbą sortowanej kolumny liczonej od 0;

KP — koniec sortowanego pola, czyli numer pierwszego niesortowanego znaku (liczony od 0), dla pól liczbowych jest to więc (TS+1)\*6.

Program rozpoznaje sposób sortowania (liczbowy czy znakowy) według dwóch bajtów. Ich określenie jest pokazane w programie demonstracyjnym:

M\$(136,136)=CHR\$(14) — dla sortowania liczbowego

=CHR\$(30) — dla sortowania znakowego

M\$(174,174)=CHR\$(7) — dla sortowania liczbowego

=CHR\$(11) — dla sortowania znakowego

Listing programu jest wydrukowany z kodami kontrolnymi "Edytora Basica" („Bajtek” 1/90). Dla uzyskania znaków w cudzysłowach należy w wierszu 60 użyć <CONTROL>N oraz <CONTROL>G, a w wierszu 240 <ESC> i <CONTROL>+ oraz <CONTROL>K.

Petr Váalka

```
TR 10 REM *** SORT 2.0 ***
YS 15 REM Ing. Petr Váalka
MX 20 REM (c) 1990, Bajtek
TN 30 ? "N":GOSUB 30010
UM 40 ? "Tablica Liczb/Znakow?":GOSUB 19
JJ 0:IF ODP=A5C("Z") THEN 230
IT 50 REM ----- SORTOWANIE LICZB -----
60 ? "Liczb kolumn tablicy":INPUT S
L:SL=SL-1:M$(136,136)="":M$(174,174)=""/>

```

# PIĘKNO MATEMATYKI

**Interesuję się matematyką i dzięki komputerowi mam tu wielkie pole do popisu.**

Ostatnio szczególnie zainteresowały mnie wzory rekurencyjne i związane z nimi grafiki. Jest to temat bardzo pasjonujący. Korzystając z jednego algorytmu, poprzez zmianę jakiejś stałej, otrzymujemy niepowtarzalne obrazy.

Chciałbym czytelnikom „Bajtka” zaprezentować dwa krótkie programy ukazujące piękno grafiki komputerowej, uzyskanej za pomocą wzorów rekurencyjnych (uzależniających współrzędne punktu następnego od współrzędnych punktu bezpośrednio go poprzedzającego). Programy są podane w najprostszej formie.

Pierwszy program poprzedzony jest deklaracjami zmiennych wykorzystywanych we wzorze, gdzie S jest skalą powiększenia, a X1 i X2 to współrzędne określające położenie figury na ekranie (pozostałe są dowolne). Dalej następuje wybór trybu graficznego i kolorów. Teraz rozpoczyna się główna pętla programu — oblicza ona współrzędne początkowe rysowanego punktu. Następnie program sprawdza, czy punkt mieści się na ekranie (bez tego nie można uzyskać powiększeń) i rysuje go. Później obliczane są nowe współrzędne i pętla jest powtarzana.

Algorytm jest wykonywany w nieskończoność, można go przerwać tylko klawiszem <BREAK> lub <RESET>. Oto kilka wartości zmiennych, dla których po upływie około 10 minut można otrzymać różne ciekawe desenie (dla  $S=1$ ):

A	B	C	D
-9	5	0.3	4
82	15	-5	0.03
242	-15	-5	0.03
10	-10	-10	4
2	-0.5	0.4	10
0.6	-8.5	0.4	7

Drugi program działa na innej zasadzie. Oparty jest na dwóch pętlach tworzących sobą wymiary rysowanego kwadratu — mozaiki, bo efektem obliczeń jest interesująca mozaika. Znow — początek to deklaracje zmiennych i wybór trybu grafiki oraz kolorów. Dalej rozpoczynają się pętlice liczące pomocnicze zmienne X i Y, potrzebne do określenia C. Punkt rysowany jest, gdy C jest niepodzielne przez dwa. Zamiast dzielenia przez 2 można dzielić przez inną liczbę i np. uzależnić od niej kolor rysowanego punktu. Uzyskamy wtedy ciekawy różnokolorowy wzór geometryczny (oczywiście trzeba wtedy zmienić tryb graficzny).

*Jacek Pliszczyński*

```

CK 1 A=-9:B=5:C=0.3:D=4:S=1:X1=160:X2=90
VI 2 GRAPHICS 24:SETCOLOR 1,0,0:SETCOLOR
    2,15,15:COLOR 1
IH 3 XP=(X*D+X1)*S:YP=(Y*D+X2)*S
WE 4 IF XP>0 AND XP<319 AND YP>0 AND YP<1
    91 THEN PLOT XP,YP
DG 5 XX=Y-SGN(X)*SQR(ABS(B+X-C)):YY=A-X:Y
    =YY:X=XX:GOTO 3
    
```

```

DZ 1 A=-15:B=-20:S=87:GRAPHICS 7:COLOR 1
DZ 2 FOR I=10 TO 100:FOR J=1 TO 80
OT 3 X=A+(S*I/100):Y=B+(S*J/100):C=INT(X*
    X+Y*Y)
UZ 4 IF C/2<>INT(C/2) THEN PLOT I,J
BB 5 NEXT J:NEXT I
    
```

## MOJE Atari

to pismo użytkowników Atari  
Jeśli masz Atari i nie chcesz zostać w tyle,  
szukaj pisma MOJE ATARI

Wydawca: Spółdzielnia „Bajtek” Warszawa, ul. Wspólna 61



### czyli miesiąc z pamiętnika Atarowca.

**Wszystkie fakty i nazwiska zawarte w tym pamiętniku są prawdziwe, a związki opisanych postaci z osobami żyjącymi nieprzypadkowe.**

#### 24.12

Pod choinką znalazłem Atari 800 XL. Jest małe, szare i dosyć sympatyczne. Całe piętro zleciało się oglądać to „magiczne pudełko”. W najbliższą sobotę jadę na giełdę kupić stację dysków.

#### 30.12

Na stację dysków nie starczyło mi pieniędzy, na magnetofon też nie, ale za to kupiłem KSO (Kasetowy System Operacyjny dla komputerów Atari współpracujący z domowym magnetofonem). Zdaniem handlarza jest on podstawowym nabytkiem do Atari, które bez niego nie może pracować. Nabyłem też kilka kaset z grami — drogie, ale za to same „nowości”.

#### 31.12

Z grania nici — KSO stosuje własny system zapisu, a kasety nagrane są standardowo. Sąsiedzi są głęboko zadowoleni, a ja z uporem maniaka próbuję wpisać prostą grę z „Bajtka”.

#### 1.1

Zarwałem noc, ale udało się! Wpisałem „Nessie” z pierwszego numeru i od samego rana rodzina i sąsiedzi myślą, jak to uruchomić. Program szczęśliwie nagrałem na kasetę, ale odczytać go już nie dałem rady. Na najbliższej giełdzie sprzedaję KSO.

#### 6.1

Widząc KSO handlarze giełdowi zdumieni pytają „co to?...”, a oferty sprzedaży komentują wymownym kółkiem na czole. Może spróbuję jeszcze raz za tydzień...

#### 13.1

Ktoś zlitował się nade mną i kupił KSO na części za 1/4 wartości. Robię

sondaz na temat magnetofonów firmowych Atari na giełdzie. Ale ceny!

#### 14.1

Po wyniesieniu z domu wszystkich książek na makulaturę, rozbięciu skarbonki swoich i rodzeństwa i długotrwałych błaganiach o wsparcie u rodziny udało mi się zebrać fundusze na stary, dwuletni XC-12. Ledwo dyszy, ale co to za radość mieć własny magnetofon.

#### 15.1

Konsternacja. Z kupionych przed tygodniem na giełdzie kaset połowa „nowości” się nie wczytuje, a reszta to kiepskie gry z 1983 roku. Mimo to nie zrażam się i w „Milipede” jestem najlepszy na piętrze. Przydałby się joystick.

#### 16.1

Mamy dość grania przy czarno-białym ekranie bez dźwięku. Rodzina kupuje przystawkę monitorową do TV za połowę ceny nowego monitora. Fachowiec instalujący przystawkę pozostawia trwały ślad u nas w domu, gdyż na perski dywan skapała cyna z lutownicy.

#### 20.1

Przystawka „trzyma się” mocno, już bez dźwięku. Fachowca wolimy więcej nie zapraszać. Zapada decyzja o kupnie monitora.

#### 25.1

Monitor spadł mi na podłogę. Po „próbie nerwów” podczas czterogodzinnego transportu środkami komunikacji miejskiej szczęśliwie docieram do punktu napraw. Tam mój sprzęt wzbudza podziw: „W jaki sposób to się jeszcze nie rozpadło?!”

#### 26.1

To po prostu niesamowite, ile racji może sprawić człowiekowi tak prosta gra jak „River Raid”. Założyłem się, że do końca tygodnia przejdę 50 poziom. Śpię trzy godziny na dobę, ale za to jaką satysfakcją!

#### 28.1

Z rekordu nici, zakład przegrałem. Na skutek przegrzania spłonął zasilacz...

#### 29.1

Monitor wrócił z naprawy, kupiłem nowy zasilacz — znowu mogę szaleć! Rozważam alternatywę zakupu stacji dysków. Może gdybym lato spędził w OHP?...

#### 30.1

Jutro wyjeżdżam na narty w Alpy. Mamusia została, ponieważ stwierdziła, że albo bierzemy ją, albo komputer. Będzie świetnie!

#### Minął rok.

#### 24.12

Pod choinką znalazłem Atari 520 ST. Jest małe, szare i dosyć sympatyczne. Całe piętro zbiegło się, by złożyć mojej rodzinie wyrazy głębokiego współczucia. W najbliższą sobotę jadę na giełdę kupić stację dysków...

**Atarowcom wszystkiego najlepszego, dużo zdrowia i satysfakcji z posiadanego sprzętu życzy autor (notabene od dwóch i pół roku szczęście właściciel Atari 65 XE).**

**Wesołej zabawy i wielu emocji przed ekranem!**

*Master*

**LICZNIK**



**CZASU**

Z zapalem czytałem zamieszczane w „Bajtku” artykuły z serii „Nie bój się przerw” i muszę przyznać, że już się przerw nie boję, a wręcz przeciwnie — są mi one bardzo pomocne w programach. Dowodem tego jest publikowany tu program.

Rola programu ogranicza się do uruchomienia procedury w języku wewnętrznym umieszczonej wcześniej na szóstej stronie pamięci. Zadaniem procedury jest odmierzenie czasu wstecz z wyświetlaniem go w lewym górnym rogu ekranu. Procedura ta może być przydatna przy pisaniu gier w

Basicu, gdyż jest wywoływana przez przerwanie TIMER2 i nie zależy od działania Basicu. Program odmierza 999 sekund, a następnie zostaje wyłączony. Odmierzany czas można skrócić po przypisaniu zmiennym X1, X2 i X3 innych wartości (z zakresu od 0 do 9) i wykonaniu instrukcji z wiersza 70. Sprawdzenie stanu licznika przez program jest możliwe i wymaga wykonania instrukcji

```
SL=(PEEK(1700)-16)*100+
(PEEK(1701)-16)*10+PEEK
(1702)-16
```

Licznik można zatrzymać i usunąć z ekranu poprzez sekwencję instrukcji

```
POKE 538, 0:GRAPHICS 0
```

Sam program jest relokowalny,

```
RB 0 REM Licznik
RF 1 REM Michal Wilga
HZ 2 REM (c) 1990, Bajtek
NI 3 REM
CL 10 X1=9:X2=9:X3=9:S=0
ZF 20 DATA 206,166,6,169,15,205,166,6,208,
,23,169,25,141,166,6,206,165,6,169,15,
205,165,6,208,8,169,25
IQ 30 DATA 141,165,6,206,164,6,169,15,205
,164,6,208,6,169,0,141,26,2,96,173,164
,6,141,64,156,173,165,6,141
QV 40 DATA 65,156,173,166,6,141,66,156,16
9,50,141,26,2,96
QA 50 RESTORE 20:FOR I=1536 TO 1605:READ
A:S=S+A:POKE I,A:NEXT I:IF S<>7521 THE
N ? "BLAD W DATA":END
XL 60 POKE 552,0:POKE 553,6:POKE 538,50
TM 70 POKE 1700,X1+16:POKE 1701,X2+16:POK
E 1702,X3+16:POKE 54286,64
```

lecz należy pamiętać, że wykonywanie on komórki 1700—1702 do przechowywania wartości cyfr. Wpisanie w to miejsce innego

programu spowoduje zaburzenia w działaniu obu programów.

*Michał Wilga*

**ZAPIS I ODCZYT RYSUNKÓW RAMBRANDTA**

Przedstawione w artykule „Polskie litery w ROM” („Bajtek” 1/89) wykorzystanie procedury ROM-RAM było bardzo proste i miało służyć tylko jako przykład.

Teraz proponuję bardziej przydatny program, który zmienia procedury systemu operacyjnego. Jest to program „LOAD/SAVE-RAMbrandt”. Dzięki niemu posiadacze magnetofonów i RAMbrandta mają możliwość nagrywania stworzonego w pocie czoła rysunku na taśmę oraz późniejszego jego odczytania. Normalna wersja kasetowa RAMbrandta nie posiada takiej funkcji.

Po przepisaniu programu za pomocą „Edytora Basicu” należy nagrać go na taśmę. Następnie trzeba wczytać jakiś program pełniący funkcję DOS-u (np. BLC, KSO, systemy operacyjne róż-

nych przeróbek magnetofonu), pozostawiając przy tym włączony Basic (włączyć komputer bez wciśnięcia klawisza <OPTION>). Po zgłoszeniu się Basicu wczytujemy „LOAD/SAVE-RAMbrandt” i uruchamiamy go. Powinna się zgłosić wczytana wcześniej namiastka DOS-u. Teraz postępujemy konwencjonalnie. Za pomocą odpowiedniej funkcji odczytujemy RAMbrandt i możemy już rysować.

Funkcję nagrywania na taśmę zawartości ekranu uzyskujemy przez naciśnięcie klawisza <HELP>, a następnie <SELECT>. Gdy chcemy wczytać obrazek, to po <HELP> naciskamy klawisz <START>. Należy uważać przy zmianach trybu graficznego z O na inne, gdyż program nie sprawdza wielkości pamięci obrazu.

**UWAGA:** Większość kasetowych kopii RAMbrandta jest zapisana jako pliki wczytujące się automatycznie (BOOT) i należy je najpierw przerobić na pliki DOS-u (inny nagłówek i inny sposób urucha-

miania). Można tego dokonać przy pomocy różnych programów kopiujących, np. Super Copy. Uwaga ta nie dotyczy systemów mogących uruchamiać takie pliki (np. TURBO ROM).

Zasada działania mojej procedury jest stosunkowo prosta. Po uruchomieniu przepisuje ona ROM do pokrywającego ją obszaru RAM. Następnie dokonuje zmiany procedury przerwania synchronizacji pionowej, wpisując pod adresem 49539 zamiast rozkazu JSR 48749 skok do własnej procedury JMP 1536, po czym wyłącza Basic i przechodzi do DOS-u. Od tej chwili naciśnięcie podanych wcześniej klawiszy wywołuje nagrywanie lub odczytywanie obrazu. Ingerencja w ROM była konieczna, gdyż chciałem, aby moja procedura działała z innymi programami, które korzystają z przerw synchronizacji pionowej.

*Sebastian Siwy*

```
QV 0 REM LOAD/SAVE - RAMBRANDT
OJ 1 REM Sebastian Siwy
HZ 2 REM (c) 1990, Bajtek
NI 3 REM
NN 10 GOSUB 1030
MV 15 REM WYLACZENIE BASICA
FS 20 RESTORE 30:FOR I=35000 TO 35012:REA
D A:POKE I,A:NEXT I
FV 30 DATA 169,192,133,106,169,254,141,1,
211,108,10,0,96
QM 35 REM PROCEDURA "DO ROM"
MO 40 S=0:RESTORE 50:FOR I=52224 TO 52372
:READ A:POKE I,A:S=S+A:NEXT I
VV 45 IF S<>16479 THEN ? "BLAD W PROCEDUR
ZE 'DO ROM':":END
HQ 50 DATA 72,138,72,152,72,173,220,2,201
,17,208,13,162,64
CR 55 DATA 173,31,208,201,6,240,102,201,5
,240,111,104,168,104,170,104,76,85,194
,169
EL 60 DATA 0,141,220,2,162,84,169,3,157,6
6,3,169,0,157,68,3,169,205,157,69
YT 65 DATA 3,169,20,157,72,3,169,0,157,73
,3,169,255,157,75,3,32,86,228,173
NL 70 DATA 255,204,157,66,3,165,88,157,68
,3,165,89,157,69,3,169,0,157,72,3
AC 75 DATA 169,30,157,73,3,32,86,228,169,
12,157,66,3,169,0,141,220,2,32,86
```

```
LF 80 DATA 228,169,255,141,14,212,24,144,
158,169,4,157,74,3,169,7,141,255,204,2
4
UM 85 DATA 144,153,169,8,157,74,3,169,11,
141,255,204,24,144,140
KM 90 POKE 52480,ASC("C"):REM POD ADRESEM
52480 WPISAC NAZWE URZADZENIA
DB 95 POKE 54286,0:POKE 49540,0:POKE 4954
1,204:POKE 54286,255:A=USR(35000):END
KA 1029 REM *** PROCEDURA "ROMRAM" ***
- Joe Miller "MAPPING THE ATARI"
YI 1030 RESTORE 1050:S=0:FOR I=1536 TO 16
35:READ A:POKE I,A:S=S+A:NEXT I:IF S<>
16212 THEN ? "BLAD W 'ROMRAM':":END
HQ 1040 A=USR(1536):RETURN
DN 1050 DATA 169,0,133,203,133,205,169,19
2,133,204,169,64,133,206,160,0,177,203
,145,205,200,208,249,230,206,230
UY 1060 DATA 204,240,12,165,204,201,208,2
08,237,169,216,133,204,208,231,8,120,1
73,14,212,72,169,0,141,14,212
IP 1070 DATA 173,1,211,41,254,141,1,211,1
69,192,133,206,169,64,133,204,177,203,
145,205,200,208,249,230,204,230
WA 1080 DATA 206,240,12,165,206,201,208,2
08,237,169,216,133,206,208,231,104,141
,14,212,40,104,96
```

# KLAN COMMODORE PIERWSZE KROKI Z COMMODORE

**Komputery są nadal dla wielu ludzi czarną magią. Otrzymujemy setki listów informujących o nieznanym ogólnych zasadach działania komputera, a co gorsza — o nieumiejętności uruchomienia go. Dlatego też zamieszczamy w tym numerze kilka podstawowych informacji: jak włączyć, wgrać program, współpracować itp.**

## JAK PODŁĄCZYĆ?

Globalnie zasady te są podobne dla wszystkich komputerów typu Commodore. Komputer wymaga prądu, trzeba więc podłączyć go do zasilacza. Jeden jego koniec wkładamy do kontaktu, a drugi do gniazda (POWER).

Drugą niezbędną rzeczą jest monitor lub telewizor. Ten pierwszy wymaga kabla monitorowego (zwykły monofoniczny kabel używany do przegrywania). Podłączany jest on do gniazda (VIDEO). Kabel telewizyjny natomiast składa się z dwóch okrągłych wtyków. Ten z bolcem wkładamy do komputera w port TV (RF), drugi koniec wsuwamy w odpowiednie gniazdo w telewizorze. Należy się jednak liczyć z tym, że telewizor bez systemu PAL/SECAM nie będzie generował kolorów i dźwięków.

Po wykonaniu tych czynności i przełączeniu włącznika w pozycję ON ukazuje się system. W konfi-

guracji monitor-komputer dostępny jest tylko Basic, różny dla różnych typów Commodore. Poza tym nie możemy praktycznie wykorzystać pełnych możliwości komputera, ani nawet raczyć się widokiem licznych gier. W tej sytuacji aż prosi się, by podłączyć jakieś urządzenie peryferyjne (magnetofon lub stację dysków), które umożliwi ładowanie programów.

## MAGNETOFON

Jest to dość tanie (30-40 \$) i przez to najbardziej rozpowszechnione urządzenie peryferyjne do Commodore. Firma Commodore produkuje tylko jeden typ tego magnetofonu (1531), który jest przeznaczony głównie dla C16 i C116. Odpowiednia przejściówka przysposabia jednak go do C64 i C128. Magnetofon podłączamy do portu (CASSETTE).

Programy zapisywane są na taśmie w tzw. turbo. Wymaga to wgrania najpierw programu turbo, uruchomienia go, a dopiero potem załadowania właściwej gry.

Bezpośrednio z Basica ładujemy instrukcją LOAD „nazwa programu”. Najbardziej rozpowszechnione turba wymagają sekwencji LOAD „np” lub ←L”np”.

Podczas ładowania może zdarzyć się kilka podstawowych błędów:

- magnetofon ma źle ustawioną głowicę — program nie wgra się lub ładuje się z błędem (? LOAD ERROR) — trzeba odpowiednio zmienić skos głowicy;
- program jest przerwany lub źle skopiowany. Ukazuje się informacja (? LOAD ERROR) lub też pro-

gram po komendzie RUN nie uruchomi się. W tym wypadku należy ponownie skopiować program;

— przekaz danych zostanie zakłócony (poruszenie magnetofonem, wciśnięcie przycisku STOP) — ładowanie należy powtórzyć.

Jeśli program wgra się bez błędów, uruchamiamy go instrukcją RUN. Większość gier wymaga użycia joysticka w pierwszym lub drugim porcie (CONTROL PORT 1-2). Nie muszą chyba wspominać, że joystick powinien mieć odpowiedni do danego typu komputera wtyk.

## STACJA DYSKÓW

Standard Commodore wymaga z reguły dysków 5 1/4 cala. Do nich przystosowane są stacje 1540, 1541, 1570, 1571. Najnowsze produkty uwzględniały też nośniki 3 1/2 cala — stacje 1581.

Stację dysków podłączamy w komputerze do portu (SERIAL), a w stacji dysków do jednego z dwóch wejść. Do drugiego z nich możemy podłączyć inną stację lub np. drukarkę czy ploter. Kable używane w tych urządzeniach są identyczne.

Informacje o zawartości dyskietki odczytujemy w następujący sposób:  
LOAD"\$",8

LIST

Po komendzie LIST ukazuje się katalog dyskietki. W niektórych typach Commodore możemy też odczytać go komendą DIR lub DIRECTORY. Efekt będzie taki sam.

Jeśli chcemy wgrać dany program z dyskietki, należy wykonać następującą sekwencję:  
LOAD „nazwa programu”, 8 lub 8,1.

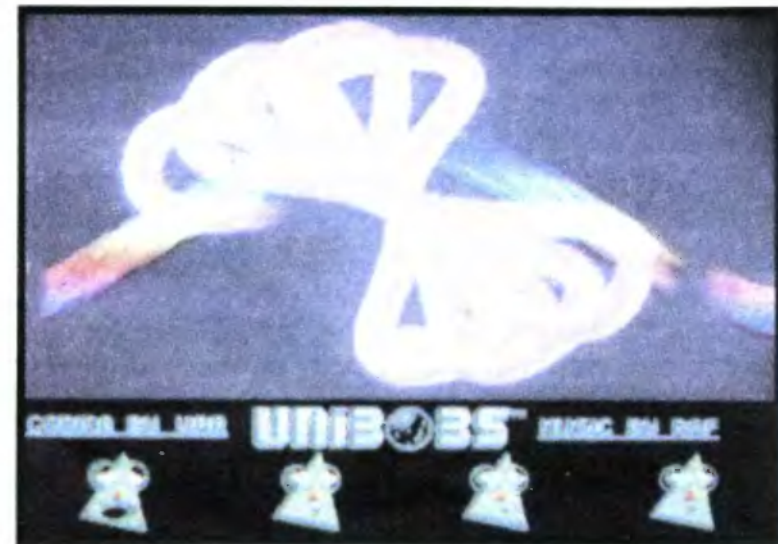
W przypadku gdy komputer nie znajduje odpowiedniego pliku, pisze informację (? FILE NOT FOUND ERROR). Należy wtedy jeszcze raz dokładnie sprawdzić katalog dyskietki i poprawić błąd.

Stacja dysków posiada z reguły dwie diody (czerwona i zielona) lub jedną dwukolorową. Zielona zapala się po włączeniu zasilania, a czerwona informuje o odczycie informacji z dyskietki. Jeśli podczas czytania stacja ma jakiegokolwiek kłopoty, pokazuje to mruganiem czerwonej diody. Niemożność załadowania programu sygnalizuje napis (? LOAD ERROR) lub (? BREAK ERROR).

*Łukasz Czekajewski*



# JUST FOR FUN!



**DEMO** — zazwyczaj krótki filmowy program, opatrzone efektowną grafiką i muzyką, prezentujący możliwości komputera (bądź programistów). Dla ich twórców jest to możliwość pokazania swoich umiejętności, pewien rodzaj sprawdzianu.

Wcielane są w życie (komputerowe) rozmaite, często bardzo oryginalne pomysły, będące niejako wstępem do profesjonalnego programowania. Niewątpliwie jest to również najprostszy sposób na zaistnienie w światku komputerowym, zdobycie ogólnego szacunku i poważania. Osoba taka jest czczona, wokół niej zaczynają się skupiać pomniejsi programiści, graficy, muzycy, posiadacze gotówki. Związują się grupa. Jej nazwa staje się widoczna na wielu nowych programach, wprowadzonych nieśmiertelnościach i zaczyna się szare życie hakera (pirata).

Uwieńczeniem pracy są megadema — cała — lub kilkudyskowe zbiory dem skomponowane w atrakcyjnie wyglądającą, zintegrowaną całość. Właściwie to one (MEGADEMA) są przyczyną sławy poszczególnych grup. Rozchodzą się na wszelkiego rodzaju parties (złotach), a u nas także na giełdach komputerowych. Są



„chodliwym towarem” i informacje przez nie przekazywane szybko się rozprzestrzeniają. Powstają jak grzyby po deszczu wszelkiego rodzaju biuletyny dyskowe, kwitnie szybka informacja i dezinformacja. Poszczególne grupy wiążą się w większe lub wzywają się na pojedynki. Trwa bezustan-

na wojna, tworzą i rozwiązują się różne pseudokorporacje. Zarabiają na siebie sprzedają dem, rozprzestrzenianiem pirackich kopii i w końcu pisananiem gier.

Praktycznie istnieją dwa rodzaje eksportu tych „dzieł”: listowna — stoso-

wana chyba już tylko w Polsce — oraz modemowa. Posiadanie szybkiego modemu staje się często koniecznością, by móc utrzymać się na powierzchni. Trwa wyścig...

Bez wątpliwości najwięcej dzieje się w świecie posiadaczy AMIG. Zarówno ze względu na wciąż rosnącą liczbę posiadaczy, jak i na wspaniałe możliwości, jakie daje ten komputer. Chyba każdy posiadacz AMIGI zna RED SECTOR INCORPORATION (RSI) i jego legendarne już MEGADEMO, oryginalne pomysły SCOOPEX-u, dobrze zrobione dema KERFEN-ów czy CULT-u. Pomimo chwilowego zastoju na listach rankingowych pierwsze miejsce wciąż zajmuje RSI, a także pornografia.

Nasza krajowa produkcja odrobinię pozostaje w tyle, ale i u nas powstają nowe grupy i dema. Prócz działającego prężnie QUARTETU (wydawcy „KEBABA”), ostatnim wydarzeniem jest KATHARSIS.

Krajowym producentom życzyć szczęścia.

*Laureat Złotej Klamki  
Mateusz Krauze*

# KONKURS

**Przez otwarte okno do pracowni Mistrza Komputeryka zaglądało Słońce. Dyskietyusz siedział przy małym stoliku w rogu pomieszczenia i mimo ogarniającej go senności usiłował słuchać wykładu Mistrza.**

— Każdy komputer ma w swym środku wielość bramek, logicznymi zwanych, które to bramki najprostsze jeno operacje wykonać potrafią — czy to „NIE” logiczne, czyli też „1” lubo „LUB”. Każda więc operacja, jaką twój komputer wykonywać będzie, na najdrobniejsze składniki logiczne rozbitą zostanie, a bramki, w koordynacji w takt zegara ze sobą współdziałając, dadzą jej swym logicznym sposobem radę.

Mistrz na krótką chwilę przerwał i baczny spojrzeniem obrzucił Dyskietyusza, który otworzył szeroko oczy, starając się sprawić wrażenie uważnie słuchającego.

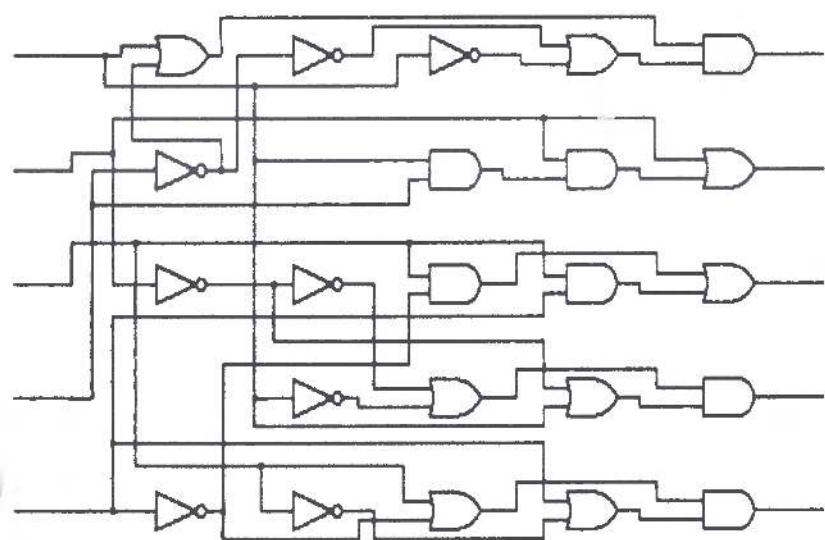
— Bramki te różnorako zaznaczać można na rycinach, schematami przez nas zwanych. Najczęstszy z tych sposobów to taki, iż bramkę „NIE” jako trójkąt z kóteczkiem się rysuje, a bramki „LUB” oraz „1” kreśli się podobnie, jeno pierwsza tył ma wklęsły i ostrze z przodu, gdy druga z tyłu płaska, a z przodu łagodnym łukiem zaokrąglona.

Dyskietyuszowi zdało się nagle, że Mistrz przestał swój wykład prowadzić, a Słońce wielką ciepłą plamą rozlało się po całym pomieszczeniu. Błogi stan nie trwał jednak długo, bo nagle pokój zatrzęsł się od głośnego krzyku Mistrza.

— Śpisz, huncwocie jeden! Nie słuchasz co mówię, boś nocą grał miast spać! Darmozjadzie, długo u mnie na terminie nie zostaniesz! Do komórki, a szybko, bo jak nie, będziesz dyski polerować!

Gwałtownie przebudzony Dyskietyusz, zastłoniwszy głowę ręką przed razami, chyłkiem wybiegł z pracowni i zbiegł po schodach do piwnicy. Po chwili zjawił się tam za nim Komputeryk. W rękę trzymał zwoje pergaminu.

— Nie myśl no, hultaju, że posiedzisz tu i wyjdiesz przed kolacją, gdy mi złość opadnie. Masz tu schemat, o jakim mówiłem, gdy ty spać się wybrałeś. Jest to schemat maszyny kodującej, którą w młodości dla Króla Choleryka Siódmego przygotował. Każdą literę według jej numeru na bity rozczłonkować trzeba było i bity przez maszynę przepuścić, im ważniejszy tym wyżej go wkładając. Z drugiej strony też jakowaś liczba z bitów pojedynczych zrobiona wychodziła, kodowana aż miło. Schematu maszyny dekodującej nie mam, bo przez lata zettał i myszy go pocięły, ale to dobrze, za łatwo ci nie będzie. Masz tu na kartce liczby, co są hasłem zakodowanym tą maszyną. Nie wyjdiesz stąd, póki hasła nie odczytasz. — Mistrz rzucił pergamin na podłogę przez Dyskietyusza i wyszedł z piwnicy, przekręcając klucz w zamku.



Maszyna kodująca Króla Choleryka III

16	9	23	3	10	26	19	10
1	12	0	7	6	21	17	19
6	1	19	6	23	28	10	1
19	0	1	29	1	22	18	19
20	17	7	5	18	19	0	1
29	1	22	18	19	6	1	19
13	26	21	19	5	29	7	0
26	18	19	5	29	1	22	18

— A tam wpadł, zupełnie jak przed rokiem! — Dyskietyusz złapał się za głowę. — Dobrze, że choć po schodach biegać nie trzeba!

Jak to wtedy z literami było? Przerwa to zero. A to jeden i tak dalej, aż do końca, do Z, które dwudzieste czwarte, bo po drodze Ł się musi zmieścić. A te bramki, co na pergaminie pozaznaczane, to pamiętam, tylko co one robią? Ta, co ją Mistrz „NIE” nazywa, to zmienia zero na jeden i na opak, „LUB” z dwóch zer robi także zero, a gdy jedynek choć jedna do niej wejdzie to jedynek wyjdzie, a ta trzecia, „1” zwana, to z niej jedynek wyjdzie, jeno wtedy, gdy do niej dwie jedynek wchodzi, inaczej zawsze zero pokazuje. Oj, może się uda chociaż przed kolacją skończyć! — I rad nierad Dyskietyusz zabrał się do pracy.

\* \* \*

Macie przed sobą kopie obu pergaminów i wiecie to samo co Dyskietyusz — spróbujcie więc rozwiązać zadanie postawione przez Mistrza Komputeryka. Jeśli będziecie mieć kłopoty, poszukajcie pomocy w książkach, chociaż hasło można odczytać na podstawie wykładu Mistrza, i tego, co Dyskietyusz pamiętał z wcześniejszych nauk.

**Spośród tych, którzy przed końcem stycznia nadeślą do redakcji na kartce pocztowej prawidłowe rozwiązanie, wylosujemy szczęśliwca, który w nagrodę dostanie komputer. Nie zapomnijcie o naklejeniu kuponu!**

# Pierwsze kroki ze Spectrum

**Jest to artykuł dla tych, którzy nigdy w życiu nie mieli do czynienia z komputerem ZX Spectrum.**

Twór Sinclaira — mały czarny komputer — posiada z tyłu obudowy cztery gniazda koncentryczną, odpowiednio oznaczone. Gniazdo **TV** służy do połączenia komputera z odbiornikiem telewizyjnym. Powinien to być odbiornik przystosowany do odbioru w zakresie **UHF**, gdyż na **36 kanale** tego zakresu pracuje nasz komputer.

Gniazdo oznaczone **9V DC** jest gniazdem zasilania i należy je podłączyć do oryginalnego zasilacza, który generuje napięcie stałe niestabilizowane 9V przy poborze prądu 1,4 A.

Pozostałe dwa gniazda umieszczone obok siebie i oznaczone **EAR** i **MIC** są gniazdami umożliwiającymi komunikację komputera z magnetofonem. Magnetofon służy jako zewnętrzna pamięć, na której możemy zapamiętywać nasze programy. Należy stale pamiętać, że wszelkie dane wprowadzone do Spectrum traci się wraz z odłączeniem komputera od zasilania.

Na tylnej ścianie ZX posiada jeszcze prostokątne wycięcie z widoczną **szyną krawędziową**. Służy ona do podłączania różnych przystawek rozszerzających możliwości komputera. Jest to najbardziej czułe miejsce nasze-

go pupila. Każde nieostrożne dotknięcie szyny krawędziowej (nawet spoconym palcem) w czasie pracy komputera może spowodować spięcie, co prawdopodobnie spowoduje uszkodzenie ważnych podzespołów.

Teraz, gdy wiemy, do czego służą poszczególne gniazda, możemy zabrać się za „odpalenie” komputerka. Należy więc:

- podłączyć komputer z odpowiednio dostrojonym telewizorem
- podłączyć odpowiednio kabelki magnetofonu; kabelek łączący gniazdo **EAR** z wyjściem słuchawkowym magnetofonu — jeśli mamy do czynienia z konwencjonalnym modelem lub z odpowiednim gniazdem też oznaczonym **EAR** dla magnetofonów przystosowanych do współpracy z komputerem. Z gniazdem **MIC** postępujemy analogicznie, lecz podłączenie go nie jest konieczne, jeśli mamy na celu tylko wgranie gier. Należy też pamiętać, że omawiane gniazda i wtyczki powinny mieć średnicę 3,5 mm
- podłączyć przystawki do złącza szufladowego (szyny krawędziowej); robimy to koniecznie przy odłączonym zasilaniu, i niech ta uwaga zapadnie Ci, Czytelniku, na zawsze w pamięć: **wszelkie operacje z szyną krawędziową**

**Spectrum dokonujemy przy wyłączonym zasilaniu komputera!**

— łączymy zasilacz z siecią (220V 50Hz) i teraz dopiero możemy wtyczkę od zasilacza ostrożnie połączyć z komputerem.

Na ekranie telewizora, w dolnej części, powinien ukazać się napis: „(c) 1982 Sinclair Research Ltd” — komputer jest gotowy do pracy.

Aby wczytać grę z kasy, należy wpisać polecenie **LOAD ""**, które oznacza, że wczytywany będzie pierwszy napotkany program na taśmie. Słowo kluczowe **LOAD** znajduje się na klawiszu **[J]**, a cudzysłów na **[P]**, lecz aby go uzyskać, trzeba wpierw nacisnąć klawisz **[SYMBOL SHIFT]**. Dopiero naciskając równocześnie te dwa klawisze otrzymamy znak cudzysłowu. Po wpisaniu tego polecenia musimy wcisnąć klawisz **[ENTER]**.

Obrzeże ekranu powinno migać, a gdy komputer odnajdzie na taśmie sekwencję początku programu, wtedy na obrzeżu pojawią się kolorowe poziome pasy, a po ok. 5 sekundach nazwa programu i ładowanie będzie przebiegać dalej. Jeśli po lub w trakcie ładowania pojawi się komunikat: **R Tape loading errorr**, należy cofnąć taśmę na początek programu i ewentualnie dostroić w magnetofonie poziom tonów wysokich lub głośność, tak aby był on odpowiedni dla komputera (zwykle wystar-

czy, aby potencjometry były w połowie skali).

Jeśli użytkownik nie używa magnetofonu kasetowego, lecz stacji dysków (TIMEX'a), powinien:

- podłączyć komputer z telewizorem jw.
- podłączyć stację z interface'em stacyjnym
- do złącza krawędziowego w Spectrum podłączyć ostrożnie interface stacyjny
- włączyć stację
- włożyć do napędu w stacji dysk z systemem
- włączyć Spectrum.

Powinien zaczytać się system z dyskietki i jeśli na dyskietce był plik **START**, to powinien się on uruchomić. W przeciwnym razie należy wprowadzić komendę **CAT\*** i po odczytaniu i wybraniu z katalogu odpowiedniego pliku należy go wgrać przez **LOAD\* „nazwa”**.

Słowo **CAT** jest dostępne w trybie EXTENDEND (**[SYMBOL SHIFT]+[CAPS SHIFT]**) po przyciśnięciu jednocześnie **[SYMBOL SHIFT]** i **[9]**. Gwiazdkę można osiągnąć przez przyciśnięcie **[B]** przy jednoczesnym trzymaniu **[SYMBOL SHIFT]**.

Przy wyłączaniu całego systemu postępujemy w odwrotnej kolejności niż przy włączaniu, należy jednak pamiętać, że podczas wyłączania stacji dysków nie powinna znajdować się w niej dyskietka.

*Brómba*

## SOUND TRACKER

**na taśmie!**

Niedługo po wersji dyskowej programu muzycznego Sound Tracker opracowana została wersja taśmowa. Współpraca z dyskiem jest opcjonalna, więc jest to wersja ulepszona. Autor, Jarosław Burczyński zapowiada kolejny program muzyczny, tym razem oparty na samlingowanych instrumentach.

Dowiedzieliśmy się niedawno, że wydany przez nas Sound Tracker na dysku został mimo zabezpieczeń złamany i jest w bezczelny sposób sprzedawany na giełdzie wraz z odbitkami oryginalnej instrukcji. Na szczęście, dzie-

ki cywilno-prawnym zabezpieczeniem, jakie poczyniliśmy, pociągnięcie piratów do odpowiedzialności jest całkowicie możliwe tym bardziej, że nie są oni nieznani. Być może w ten sposób zetniemy choć jeden kłujący chwast w dżungli polskiego pseudo-ryнку programów.

W naszej redakcji jest wciąż do nabycia autoryzowana kopia Sound Trackera, już w wersji taśma-dysk i z instrukcją. Ci, którzy kupili wcześniejszą wersję, otrzymają nową po 5% ceny.

*Redakcja*

**TOP SECRET**  
**MAGAZYN**  
**FANÓW**  
**GIER**  
**KOMPUTEROWYCH**  
**Numer drugi**  
**— znów pełen nowości,**  
**map i opisów**  
**JUŻ W KIOSKACH!**

Wydawca: Spółdzielnia „Bajtek” Warszawa, ul. Wspólna 61

# Balistyki

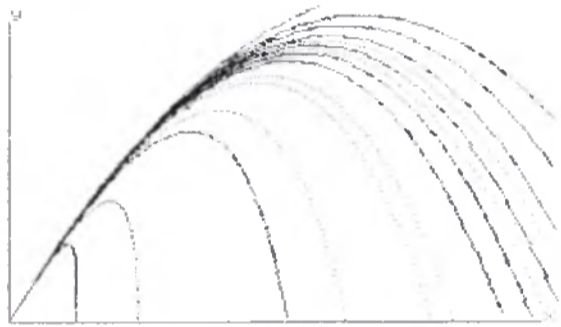
**Matrix stał na szczycie skały, która siedem centymetrów dalej urywając się odstąpiła widok na rozległą dolinę.**

W dolinie, na środku łąki, stało kilka wojskowych żaków, wśród których kręcili się niespokojnie najemnicy. Matrixa zobaczyli już pięć minut temu, co zobligowało ich do obudzenia Johna — najlepszego snajpera w północnej Dakocie. John jednym susem zeskokczył z samochodu, sprawnym ruchem załadował „fuzję”, przyłożył oko do lunety, wycelował i strzelił...

Po kolejnym niecelnym strzale Matrix zaśmiał się szyderczo. Śmiech przeszył dolinę tak, jak strzala wypuszczona z kuszy przesywa ciało nieszczęśliwca, który nie zdążył się ukryć przed śmiercią.

Nasz bohater mógł pozwolić sobie na ryzyko wystawienia się jako cel, wiedział bowiem, co to jest krzywa balistyczna i po jakim torze porusza się pocisk wystrzelony z doliny w kierunku obiektu oddalonego w pionie i poziomie.

Zagadnieniem opisanym wyżej w sfa-



Przykładowe wykresy dla różnego współczynnika oporu »b«, a tych samych składowych prędkości początkowej.

laryzowanej formie, zagadnieniem czysto fizycznym, zajmuje się **balistyka**. Jest to nauka o ruchu pocisków i jest ona podstawą w projektowaniu broni.

Krzywa balistyczna przedstawia tor pocisku, a program zamieszczony obok pozwala na zapoznanie się z charakterystycznymi cechami takiej krzywej, rysuje ją oraz pomaga w zrozumieniu zjawiska ruchu pocisku.

Po ważnym wpisaniu programu należy go nagrać na kasetę (dyskietkę) zleceniem **SAVE „BALIST” LINE 1**

Program pyta o składowe prędkości początkowej, czyli o prędkość początkową poziomą i pionową. Optymalnymi wartościami dla programu jest zakres od 10 do 120. Kolejnym pytaniem programu jest pytanie o współczynnik oporu »b«. Jest to bardzo istotne, gdyż to właśnie od oporu zależy charakterystyka toru pocisku. Jeśli współczynnik ten wynosi zero, to pocisk porusza się tak jak w próżni i jego krzywą balistyczną powinna być parabola. W naszym programie współczynnik ten nie może być większy od ośmiu, lecz może być ujemny (!) — co daje ciekawe rezultaty.

Wykres toru pocisku może być ciągły lub w postaci punktów, a zależne jest to od naszej odpowiedzi na pytanie o ciągłość wykresu.

Po narysowaniu jednej krzywej program pyta nas, czy chcemy kolejny wykres, lecz dla innego współczynnika oporu »b«. Ta opcja umożliwia nam śledzenie torów pocisku dla tych samych składowych prędkości początkowej, lecz innych oporów. Taką sytuację ilustruje rysunek.

Program jest bardzo prosty i tylko pomaga w zrozumieniu zjawiska, co czyni go odpowiednim narzędziem dla nauczyciela fizyki chcącego urozmaicić swój wykład.

*Maciej Pietras*

```

10 BEEP 0.6,41: PAPER 7: OVER 0: BORDER 7: INK 0: BRIGHT 0: CLS
15 FOR J=64 TO 71: POKE 23681,J: LPRINT INVERSE 1;"
   F I Z Y K A          ": NEXT J:    REM to sztuczka z po
wieszaniem napisow
17 PRINT AT 10,6;"PARABOLE i BALISTYKI"
18 BEEP .1,20: POKE 23658,0
20 PRINT #0; INK 2; INVERSE 1;TAB 1;"naclisnij ENTER dla ko
ntynuacji "
25 IF CODE INKEY#<>13 THEN GO TO 25
28 REM program glowny
29 CLS
30 LET dt=.1: REM krok calkowania
40 LET xstare=0: LET ystare=0
45 LET draw=0
50 LET vx=0: LET vy=0:           LET g=9.81
70 PRINT AT 20,0;"Podaj wartosc skladowych          predkosci
i poczatkowej:"
80 LET skala=0: INPUT "Vx= ";vx;"Vy= ";vy: LET uxs=vxs:
LET uys=vys
82 IF vxs>120 THEN LET vxs=vxs/3: LET vys=vys/3: LET uxs=
vxs: LET uys=vys: LET skala=1
90 CLS : PRINT AT 20,0;"Podaj wartosc wspolczynnika ""oporu
">>b<< "
92 INPUT "b= ";b: IF b>8 THEN BEEP .1,-20: PRINT #0;"ZA D
UZE >>b<<": PAUSE 0: GO TO 92
95 CLS : GO SUB 600
97 CLS : PLOT 0,0: DRAW 255,0: PLOT 0,0: DRAW 0.175: PRINT
AT 21,31;"x";AT 0,0;"y"
98 IF skala=1 THEN PRINT AT 0,22: FLASH 1;"skalowanie"
100 FOR q=0 TO 100000000
110 LET ax=-1*b*vxn
120 LET vx= vx+ax*dt
130 LET wx=(vx+vxs)/2
140 LET xnowe=xstare+wx*dt
150 REM
160 LET ay=-1*g-b*vy
170 LET vy= vy+ay*dt
180 LET wy=(vy+vys)/2
190 LET ynowe=ystare+wy*dt
200 IF ynowe<0 THEN GO TO 400
205 IF ynowe>175 THEN GO TO 400
210 IF xnowe>=255 THEN GO TO 400
240 PLOT xnowe,ynowe
245 IF draw=1 THEN DRAW -(xnowe-xstare),-(ynowe-ystare)
250 LET ystare=ynowe
260 LET xstare=xnowe
270 LET vys=vy
280 LET vxs=vx
300 NEXT q
400 PRINT #0;"Jeszcze jeden wykres ? (t/n)": PAUSE 0: IF IN
KEY#="n" THEN CLS : GO TO 10
410 GO SUB 600
450 LET xstare=0: LET ystare=0: LET vx=0: LET vy=0: LET y
nowe=0: LET xnowe=0: LET vys=uy: LET vxs=ux: INPUT "Wspolc
zynnik oporu >>b<< ";b
460 IF b>8 THEN PRINT #0;"ZA DUZE >>b<<": BEEP .1,-20: PAU
SE 0: GO TO 450
500 GO TO 100
600 INPUT ": PRINT #0;"Wykres ciagly ? (t/n)"
610 BEEP .4,-4: PAUSE 0: IF INKEY#="t" THEN LET draw=1
620 IF INKEY#="n" THEN LET draw=0
630 RETURN
700 REM
9999 SAVE "BALISTYKI" LINE 10
  
```



**Poprzednio opisaliśmy przeróbkę ZX SPECTRUM na 80 KB. Dziś prezent dla posiadaczy komputerów TIMEX 2048.**

Komputer ten posiada nieco inaczej zorganizowaną sprzętowo pamięć RAM. „Dolne” 16 KB składa się z dwu układów 4416 (inne oznaczenia: 41416, 4264, lub IMS 2620P) będących czterobitowymi pamięciami dynamicznymi 16K. Posiadają one w stosunku do pamięci 4116 stosowanych w Spectrum szereg zalet, przede wszystkim wymagają tylko jednego napięcia zasilania (+5V) i — ponieważ jedna „kostka” obsługuje od razu cztery bity szyny danych — wymagane są jedynie dwie sztuki tych pamięci. Pamięć „górna 32 KB bywa zorganizowana w dwojaki sposób: na płycie mogą być zainstalowane cztery układy 4416, co daje 2 \* 16 KB = 32 KB, lub dwa układy 4464 (lub 41464, nie mylić z układami 4264, będącymi odpowiednikami pamięci 4416) — czyli pamięci czterobitowe 64K — wykorzystywane, podobnie jak w Spectrum, w połowie.

Jeżeli na płycie mamy zainstalowane (pod oznaczeniami U8, U9, U10 i U11) układy 4416 lub odpowiedniki, to musimy zakupić dwa układy początkowych pamięci 64 K, (4464, 41464, 82464 itp.) i dokonać wymiany pamięci. Jeśli miejsca na układy

U10 i U11 są puste, a jako U8 i U9 mamy na płycie od razu wlutowane układy 64 K, to odpadnie nam ich zakup i wymiana. Takı przypadek jest jednak rzadszy.

Układ elektroniczny przełączania banków pamięci jest identyczny jak dla ZX Spectrum (patrz poprzedni numer „Bajtki”). Także płytkę drukowaną wykonujemy według zamieszczonego tam rysunku. Jednakże podłączenie jej do komputera oraz ewentualną wymianę pamięci wykonujemy nieco inaczej.

Wymiana pamięci przebiega w trzech etapach. Najpierw ostrożnie wylutowujemy układy U8, U9, U10 i U11. Po sprawdzeniu, czy nie uszkodziliśmy ścieżek na płycie, i po oczyszczeniu otworów należy wlutować w miejsce układów U8 i U9 pamięci 4464. Następnie należy usunąć zwróć wlutowaną zamiast diody D22, która znajduje się na prawo od układu pamięci U7. W miejsce zwróć (oznaczone D22) i w miejsce tuż obok (oznaczone D23) wlutowujemy dwie diody krzemowe impulsowe, np. BAVP 17, 18, ... 12, BAYP 95, 1 N 4148 itp. W miejsce oznaczone R64 wlutowujemy rezystor 4,7 K. Po sprawdzeniu, czy nie popełniliśmy błędów lub nie spowodowaliśmy drobnego, ale przykrego zwarcia na płycie, wkładamy komputer do obudowy i włączamy go. Jeśli zgłosi się on znajomym komunikatem i na polecenie PRINT USP „a” otrzymamy odpowiedź 65368, to pamięci zostały zamontowane poprawnie. Jeśli nie — natychmiast wyłączamy komputer i szukamy błędów.

Po ewentualnej wymianie pamięci i zmontowaniu na płycie drukowanej układu przełączania banków przystępujemy do jego instalacji w komputerze. Sygnały adresów i sterujące podłączamy tak jak w Spectrum: do odpowiednich punktów lutowniczych w okolicy złącza krawędziowego lub procesora. Pozostaje sprawa podłączenia sygnału wybierającego

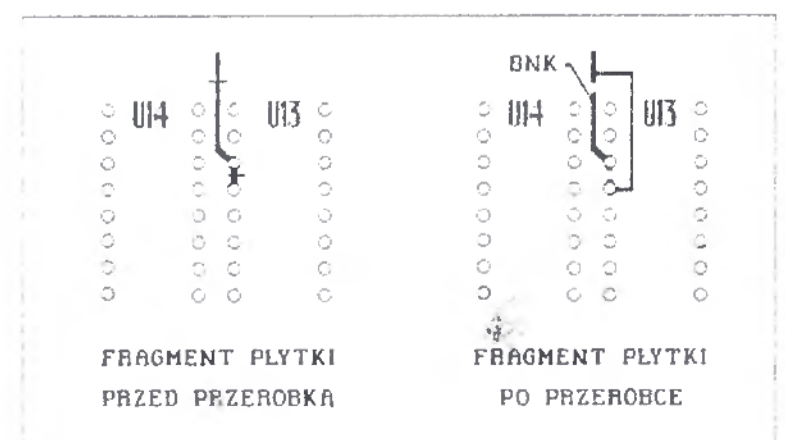
bank pamięci „BNK”. Ponieważ w Timex-ie nie ma zwróć ustalających wykorzystywaną połowę układów 64K, musimy „podłączyć się” bezpośrednio do multiplexera adresów. Należy zlokalizować ścieżkę biegnącą do 11 i 12 nóżki układu U13 (typu 74LS157). Znajduje się ona na dolnej stronie płytki komputera (patrz rysunek). Ścieżkę tę należy przeciąć ostrym nożem w dwu zaznaczonych na rysunku miejscach. Następnie łączymy odciętą ścieżkę bezpośrednio z nóżką 11 układu, do nóżki 12 zaś doprowadzamy sygnał „BNK” z przełącznika banków. I to już wszystko.

Wszelkie uwagi dotyczące uruchamiania tej przeróbki podane w poprzednim artykule pozostają aktualne. Należy zwrócić uwagę, by nie spowodować przypadkowych zwarcí podczas dokonywania przeróbki płytki oraz upewnić się, że dokładnie przecięliśmy ścieżki.

Zyczymy powodzenia i wielu pomysłów zastosowania rozszerzonej pamięci.

*Stanisław Winiecki*

**Autor dziękuje kol. Krzysztofowi Teofilakowi za udostępnienie komputera w celu wykonania układu modelowego.**



# MAGAZYNIER

```

10 GO SUB 3000: GO SUB 4000: GO TO 1000
40 REM Czekanie na Klawisz
50 IF INKEY$<>" " THEN GO TO 50
60 IF INKEY$=" " THEN GO TO 60
70 LET k$=INKEY$: RETURN
100 REM czytanie klawiatury
110 LET kx=0: LET ky=0: GO SUB 40
140 IF k$=m$ THEN LET ky=1
150 IF k$=n$ THEN LET kx=1
160 IF k$=o$ THEN LET ky=-1
170 IF k$=p$ THEN LET kx=-1
180 RETURN
200 REM czytanie joysticka - Kempston
210 LET kx=0: LET ky=0: LET k$=INKEY$
220 LET j=IN 31: IF j>15 THEN LET j=j-16
230 IF j>7 THEN LET ky=-1: GO TO 270
240 IF j>3 THEN LET ky=1: GO TO 270
250 IF j>1 THEN LET kx=-1: GO TO 270
260 IF j THEN LET kx=1
270 RETURN
300 REM ruch magazyniera
310 PRINT AT nym, nym; "E"; AT pym, pxm; d$(pym, pxm)
320 LET pxm=nym: LET pym=nym: LET kr=kr+1
330 RETURN
1000 REM petla glowna programu
1010 PRINT AT 20, 0; "kroki "; kr, "pchnięcia "; pch
1020 GO SUB ruch: IF k$="O" THEN GO TO 1200
1030 IF kx=0 AND ky=0 THEN GO TO 1020
1040 LET nxm=pxm+kx: LET nym=pym+ky: LET a$=d$(nym, nxm)
1050 IF a$="B" OR a$="C" THEN GO SUB 300
1060 IF a$<>"A" THEN GO TO 1160
1070 LET nyp=nym+ky: LET nxp=nxm+kx: LET b$=d$(nyp, nxp)
1080 IF b$<>"C" AND b$<>"B" THEN GO TO 1160
1090 PRINT AT nyp, nyp; "A"
1100 GO SUB 300: LET d$(nym, nxm)="B"
1110 IF e$(nym, nxm)="C" THEN LET d$(nym, nxm)="C"
1120 LET d$(nyp, nxp)="A"
1130 IF b$="C" THEN LET pcz=pcz-1
1140 IF e$(nym, nxm)="C" THEN LET pcz=pcz+1
1150 LET pch=pch+1
1160 IF pcz>0 THEN GO TO 1000
1170 CLS: PRINT AT 3, 3; "Brawo! Udało Ci się zrobić"
1180 PRINT AT 6, 3; "Porządek w magazynie nr "; mag; "."
1182 IF mag<nmag THEN GO TO 1185
1183 CLS : PRINT AT 8, 7; "Skończyłeś grę!!!"
1184 PRINT AT 14, 10; "GRATULACJE!": STOP
1185 LET mag=mag+1: PAUSE 100: GO SUB 4040: GO TO 1000
1200 CLS : PRINT AT 3, 2; "Niestety, w magazynie został"
1210 PRINT AT 6, 2; "bałagan."
1230 PRINT AT 15, 4; "Czy grasz dalej? (T/N)"
1240 GO SUB 40
1250 IF k$<>"t" AND k$<>"T" THEN STOP
1260 GO SUB 4000
1270 GO TO 1000
3000 REM opis, UDG
3010 BORDER 5: PAPER 5: CLS
3020 PRINT AT 3, 10; FLASH 1; "Magazynier"
3030 PRINT AT 6, 0; "wg Soko Ban napisał M. Borkowski"
3040 PRINT "Ułoż wszystkie paczki na swoich miejscach."
3050 PRINT "Paczki przesuwaj popychając je"
3060 PRINT "Jeśli nie możesz skończyć gry"
3065 PRINT "naciśnij klawisz " "0" " ."
3070 DIM d$(16, 20): DIM e$(16, 20)
3080 FOR i=USR "a" TO USR "e"+7
3090 READ a: POKE 1, a
3100 NEXT i
3110 PRINT AT 20, 4; "Naciśnij dowolny klawisz. ": GO SUB 40
3120 REM wybór klawisze-joystick
3130 CLS : PRINT AT 3, 1; "Jak kierujesz magazynierem?"
3140 PRINT AT 8, 9; BRIGHT 1; "j"; BRIGHT 0; "oystick"
3150 PRINT AT 12, 9; BRIGHT 1; "k"; BRIGHT 0; "lawiatura"
3160 GO SUB 40: CLS
3170 IF k$="j" OR k$="J" THEN LET ruch=200: GO TO 3280
3180 REM definiowanie klawiszy
3190 PRINT AT 5, 5; "Naciśnij klawisz LEWO"
3200 GO SUB 40: LET p$=k$
3210 PRINT AT 5, 22; "PRAWO": GO SUB 40: LET n$=k$
3220 PRINT AT 5, 22; "GORA ": GO SUB 40: LET o$=k$
3230 PRINT AT 5, 22; "DOŁ ": GO SUB 40: LET m$=k$
3270 LET ruch=100: CLS
3280 RETURN
3500 REM dane do UDG
3510 DATA 0, 30, 38, 122, 74, 76, 120, 0
3520 DATA 0, 0, 0, 0, 0, 0, 0, 0
3530 DATA 0, 24, 36, 66, 66, 36, 24, 0
3540 DATA 187, 187, 187, 0, 221, 221, 221, 0
3550 DATA 56, 56, 16, 124, 124, 124, 40, 40
4000 REM wczytanie deski
4010 RESTORE 4990: READ nmag
4020 INPUT ("Który magazyn (jest "; nmag; ") "); mag
4030 IF mag>nmag THEN GO TO 4020
4040 RESTORE 5000+(20*(mag-1))
4050 BORDER 5: PAPER 5: CLS : PRINT : PAPER 6
4060 FOR i=1 TO 16
4070 READ d$(i): LET e$(i)=d$(i): PRINT " "; d$(i)
4080 NEXT i
4090 READ pxm, pym: LET pxm=pxm+1: LET pym=pym+1
4095 PRINT AT pym, pxm; "E"
4100 READ pcz: LET kr=0: LET pch=0
4110 PRINT AT 0, 4: PAPER 5; "Magazyn nr "; mag
4120 RETURN
4990 DATA 5: REM ilość magazynów

```

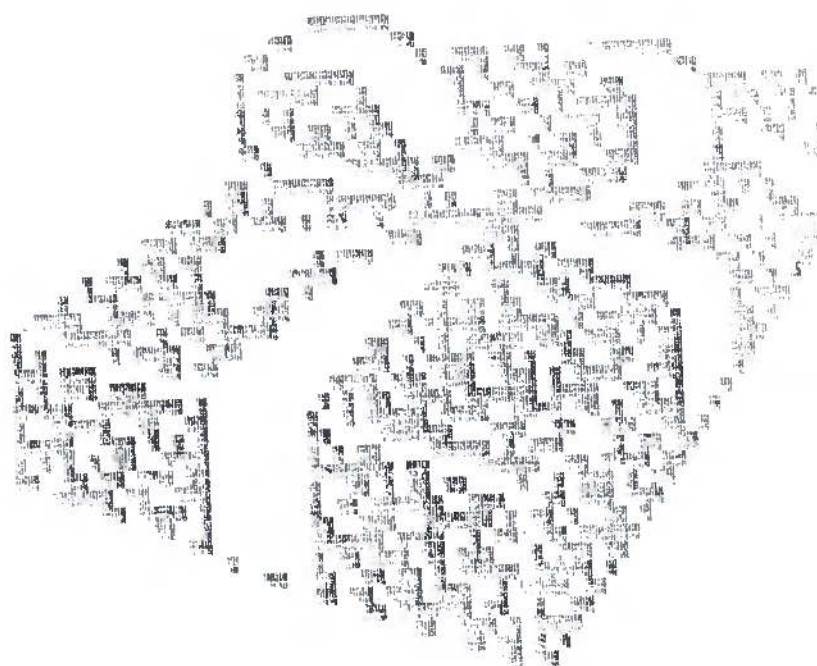
```

4995 REM dane do magazynów
5000 DATA "BBBBBBBBBBBBBBBBBBBB"
5001 DATA "DBBBBBBBBBBBBBBBBB"
5002 DATA "DBAABBAABBAABCCCD"
5003 DATA "DBBAAADBBBABBCCCD"
5004 DATA "DBABBBBAABAAABCCCD"
5005 DATA "DDBBBBBBAABBBBCCCD"
5006 DATA "DBBBBBBBAABABDCCCD"
5007 DATA "DBBBBBBBBBBDDCCCD"
5008 DATA "DBDBBBBBAABABDCCCD"
5009 DATA "DBBDBBBAABABADCCCD"
5010 DATA "DBCCDDBBBAABBBBCCD"
5011 DATA "DBCCDDBBAAABAAABCCD"
5012 DATA "DDDDDBBBBBBBBDDCCD"
5013 DATA "BBBBBDDDDDDDDDBCCD"
5014 DATA "BBBBBDDDDDDDDDBCCD"
5015 DATA "BBBBBDDDDDDDDDDDD"
5016 DATA 2, 1, 32
5020 DATA "BBBBBBBBBBBBBBBBBBBB"
5021 DATA "BBBBBBBBBBDDDDBBBB"
5022 DATA "BBBBDDDDDBBBBDBBB"
5023 DATA "BBDDDBDDDAABDBBB"
5024 DATA "BBDDBBBBBAABDBBB"
5025 DATA "BDDDBAABADDDDBBB"
5026 DATA "BDDADDBBBBBDBBB"
5027 DATA "BDDDBAABADDDDBBB"
5028 DATA "BDBBABBDBBBDDBDD"
5029 DATA "DDDDBBBBBBAABDBBB"
5030 DATA "DDDDDBBBAABBBBDB"
5031 DATA "DCBBBBDDDDDDDDDD"
5032 DATA "DCCBCCDDDDDDDBBB"
5033 DATA "DCCCDDBBBBBBBBBB"
5034 DATA "DCCCDDBBBBBBBBBB"
5035 DATA "DDDDDBBBBBBBBBBB"
5036 DATA 7, 4, 14
5040 DATA "BBBBBBBBBBBBBBBBBBBB"
5041 DATA "BBBBBBBBBBBBBBBBBBBB"
5042 DATA "BBBBBBBBBBBBBBBBBBBB"
5043 DATA "BDBBBBBBBBBBBBBB"
5044 DATA "BDBDDDDDDDBBBBDB"
5045 DATA "BDBBBAABABADDBBB"
5046 DATA "BDBBBAABABBDDBB"
5047 DATA "BDBBDAABABDDCCCD"
5048 DATA "BDBBBAABABDCCCD"
5049 DATA "BDDDDAAABABDDCCCD"
5050 DATA "BDBBBBBBDDDDCCCD"
5051 DATA "BDDDDDBBDDDDCCCD"
5052 DATA "BBBBDDDDDBBBBDB"
5053 DATA "BBBBBDDDBBBBBB"
5054 DATA "BBBBBDDDDDDDBBB"
5055 DATA "BBBBBBBBBBBBBBBBBBBB"
5056 DATA 8, 6, 15
5060 DATA "BBBBBBBBBBBBBBBBBBBB"
5061 DATA "BBBBDDDDDDDBBBB"
5062 DATA "BDDBBBDDDDDBBB"
5063 DATA "DDDBBBBBBDBBBDD"
5064 DATA "DBBBDABDBBDBCCCD"
5065 DATA "DBBBDABDBBDDCCCD"
5066 DATA "DBBBDABDBBDBCCCD"
5067 DATA "DBABBBABDBBDDCCCD"
5068 DATA "DBBBDDBBDAABCCCD"
5069 DATA "DBABBBBDBBDDCCCD"
5070 DATA "DBBABBABBBABCCCD"
5071 DATA "BDBDDDDDBBBBDB"
5072 DATA "BDBBDDDDDDDDDB"
5073 DATA "BDDDDBBBBBBBBB"
5074 DATA "BBBBBBBBBBBBBBBBBBBB"
5075 DATA "BBBBBBBBBBBBBBBBBBBB"
5076 DATA 6, 5, 16
5080 DATA "BBBBBBDDDDDBBB"
5081 DATA "BDDDDDDDBBBBDB"
5082 DATA "BDBBBBBBBAABDB"
5083 DATA "BDBAABBBDDDDDD"
5084 DATA "BDBDDDDCCCCDDDB"
5085 DATA "BDBBBBCCCCDDDB"
5086 DATA "BDBDDDDCCCCDBB"
5087 DATA "BDBBDDDDDBDDAD"
5088 DATA "BDBBDABBBBABBDB"
5089 DATA "BDBBBAABABDDBDB"
5090 DATA "BDBBBAABDDDAABDB"
5091 DATA "BDDDDBBBBBABBDB"
5092 DATA "BBBBBDDDBBDBB"
5093 DATA "BBBBBDDDBBDBB"
5094 DATA "BBBBBDDDDDBBDB"
5095 DATA "BBBBBDDDDDBBDB"
5096 DATA 11, 2, 18

```

**W** styczniowym (1-2/90) numerze „Bajtka” w klanie Atari Wojtek Zientara opublikował grę „MAGAZYNIER”, napisaną na podstawie istniejącej na komputerach klasy IBM PC gry „SokoBan”. Teraz kolejna wersja — na Spectrum. Dla wszystkich tych, którym gra się podoba, mamy niespodziankę. Otóż znajdujące się na końcu programu dane opisujące poszczególne magazyny są tak samo zorganizowane jak te, które były opublikowane w klanie Atari. Możecie więc dopisać dane ze styczniowego klanu ATARI na koniec naszej wersji (uwaga na numerację linii) i poprawić w instrukcji DATA w linii 4990 liczbę magazynów na właściwą. Podobnie mogą postąpić właściciele ATARI, którzy dali już sobie radę ze wszystkimi magazynami opublikowanymi w ich wersji gry. (Linia 4090 zapewnia zgodność formatu danych i można ją zmienić, jeśli odpowiednio zmienicie dane).  
Uwaga przy wpisywaniu programu — duże, podkreślone litery na wydruku (A,B,C,D,E) to UDG!  
Powodzenia w przesuwaniu paczek!

Marcin Borkowski





# TURBO PASCAL

## procedury graficzne na ZX Spectrum

Kompilator Turbo Pascal wersja 3.0 przeznaczony na komputery CP/M'u nie posiada wbudowanych procedur graficznych. Brakuje też ogólnie akceptowanych standardów takich bibliotek dla 8-bitowych komputerów. Każda firma ma praktycznie własną propozycję w tej dziedzinie. Problem ten dotyczy także ZX Spectrum współpracującego ze stacją FDD 3000.

Opisywany w tym artykule zbiór procedur graficznych wzorowany jest na pakiecie prezentowanym w [1] i rozszerzonym później w [2] dla komputerów Amstrad PCW. Na listingu 1 przedstawiono plik GRAPH.SYS zawierający pięć podstawowych procedur. Trzy pierwsze (PLOT, LINE i CIRCLE) w istotny sposób wykorzystują kody terminala zaproponowane przez firmę Polbrit. Procedura Ellipse jest trochę dłuższa i realizuje algorytm opisywany w [1]. Ostatnią z procedur (BOX) rysuje ramkę o współrzędnych określonych parametrami: x1,y1, x2,y2.

Listing 2 zawiera przykładowy program demonstrujący możliwości zbioru GRAPH.SYS. Efekt działania programu GraphDemo zamieszczono na rys. 1.

Jonasz Mayer

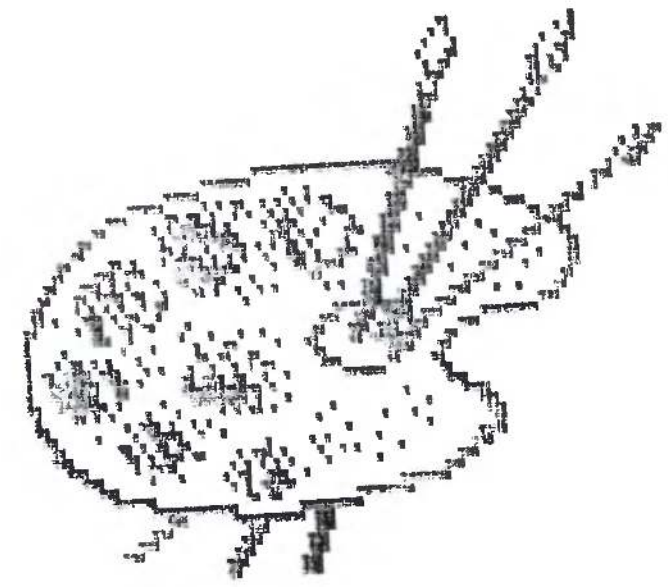
Literatura:  
[1] J. Młodzki, Algorytmy generacji krzywych na płaszczyźnie, „Bajtek” 7/88.  
[2] J. Młodzki, Minipakiet graficzny na komputer Amstrad PCW 8256/8512, „Bajtek” 8,9,12/90.

```

1: (*****)
2: (* *)
3: (*      Plik GRAPH.SYS *)
4: (*      ver. ZX.01 *)
5: (*      (C) JM 1990 *)
6: (* *)
7: (* Umożliwia realizacja podstawowych funkcji graficznych *)
8: (* w Turbo Pascalu. Do kreslenia elips zastosowano *)
9: (* efektywne algorytmy przyrostowe. *)
10: (* *)
11: (* Komputer:      ZX Spectrum + FDD 3000 *)
12: (* System operacyjny: CP/M 2.2 *)
13: (* Język:         Turbo Pascal 3.0 *)
14: (* *)
15: (* Lista procedur i funkcji: *)
16: (* *)
17: (* 1. p. PLOT   ( x,y, Pflag ; integer ) *)
18: (* 2. p. LINE   ( x1,y1, x2,y2, Pflag ; integer ) *)
19: (* 3. p. CIRCLE ( x0,y0, R, Pflag ; integer ) *)
20: (* 4. p. ELLIPSE ( x0,y0, a,b, Pflag ; integer ) *)
21: (* 5. p. BOX    ( x1,y1, x2,y2 ; integer ) *)
22: (* *)
23: (* Uwagi, *)
24: (* 1. Zmienna Pflag została wprowadzona celem *)
25: (* uzyskania zgodności z podobnym pakietem *)
26: (* procedur na komputer Amstrad PCW 8256. *)
27: (* 2. Procedury PLOT, LINE, CIRCLE wykorzystują kody *)
28: (* terminala zgodnego z terminalem Polbritu. *)
29: (* *)
30: (*****)
31:
32: procedure PLOT (x,y, PFlag; integer);
33: (*****)
34: (* procedura PLOT zapala *)
35: (* o współrzędnych x,y. *)
36: (*****)
37: begin
38:   write (#27'P',chr(Lo(x)),chr(Hi(x)),
39:         chr(Lo(y)),chr(Hi(y)));
40: end; { of Plot }
41:
42: procedure LINE ( x1,y1, x2,y2, PFlag ; integer);
43: (*****)
44: (* procedura kresli linie między para *)
45: (* punktów o współrzędnych x1,y1 a x2,y2. *)
46: (*****)
47: begin
48:   write (#27'P',chr(Lo(x1)),chr(Hi(x1)),
49:         chr(Lo(y1)),chr(Hi(y1)),
50:         #27'Q',chr(Lo(x2)),chr(Hi(x2)),
51:         chr(Lo(y2)),chr(Hi(y2)));
52: end; { of Line }
53:
54: procedure CIRCLE (x0,y0, R, PFlag ; integer);
55: (*****)
56: (* procedura kresli okrąg o środku w *)
57: (* punkcie x0,y0 i promieniu R. *)
58: (*****)
59: begin
60:   write (#27'O',chr(Lo(x0)),chr(Hi(x0)),
61:         chr(Lo(y0)),chr(Hi(y0)),
62:         chr(Lo(r)),chr(Hi(r)));
63: end; { of Circle }
64:
65: procedure ELLIPSE (x0,y0, a,b, PFlag ; integer);
66: (*****)
67: (* procedura kresli elipse o środku w *)
68: (* w punkcie x0,y0 i polosiach a,b *)
69: (*****)
70: var x,y ; integer;
71:     fa,fx,fy,fx,y, ax,ay,axy, aa,bb, a2,b2 ; real;
72: begin
73:   x := a;   y := 0;   fa := 0;
74:   aa := a;  aa := aa*aa; a2 := 2*aa;
75:   bb := b;  bb := bb*bb; b2 := 2*bb;
76:   repeat
77:     PLOT ( x0+x, y0+y, PFlag); PLOT ( x0+x, y0-y, PFlag);
78:     PLOT ( x0-x, y0+y, PFlag); PLOT ( x0-x, y0-y, PFlag);
79:     fx := fa - x * b2 + bb;   ax := abs(fx);
80:     fy := fa - y * a2 + aa;   ay := abs(fy);
81:     fxy:= fx + fy - fa;       axy:= abs(fxy);
82:     if (ax<=ay) and (ax<=axy)
83:     then begin   fa := fx;   x := x - 1; end
84:     else if (ay<ax) and (ay<=axy)
85:     then begin fa := fy;   y := y - 1; end
86:     else begin fa := fxy;  x := x - 1;
87:                y := y - 1; end;
88:   until x<0;
89: end; { of ELLIPSE }
90:
91: procedure BOX ( x1,y1, x2,y2 ; integer);
92: (*****)
93: (* procedura kresli pudełko o współrzędnych: *)
94: (* lewy górny róg - x1,y1, *)
95: (* prawy dolny róg - x2,y2. *)
96: (*****)
97: const PFlag = 1;
98: begin
99:   LINE ( x1,y1, x2,y1, PFlag );
100:  LINE ( x2,y1, x2,y2, PFlag );
101:  LINE ( x2,y2, x1,y2, PFlag );
102:  LINE ( x1,y2, x1,y1, PFlag );
103: end; { of BOX }
104:
105: (*****)
106:

```

LISTING 1. Plik GRAPH.SYS



```

Program GraphDemo;
(*****)
(* *)
(* Plik DEMO.PAS (C) JM 1988, RM 1990 *)
(* *)
(* Demonstracja możliwości graficznych *)
(* pakietu GRAPH.SYS na komputerze *)
(* ZX Spectrum, *)
(* *)
(*****)

(*$I graph.sys *)

var
  ch : char;
  i : integer;

begin { GraphDemo }
  clrscr;
  Box (0,175,255,0);
  Line (0,87,255,87,1);
  Line (85,0,85,175,1);
  Line (170,0,170,175,1);
  Line (85,44,170,44,1);

  GotoXY (1,1);   for i:=1 to 10 do write(' ');
  GotoXY (20,1); write('GRAPH DEMO (C) JM 1988');
  GotoXY (7,3);  write(' LINE ');
  GotoXY (7,14); write(' ELLIPSE ');
  GotoXY (50,3); write(' LINE ');
  GotoXY (28,3); write(' CIRCLE ');
  GotoXY (28,14); write(' PLOT ');
  GotoXY (50,14); write(' CIRCLE ');
  GotoXY (28,19); write(' BOX ');

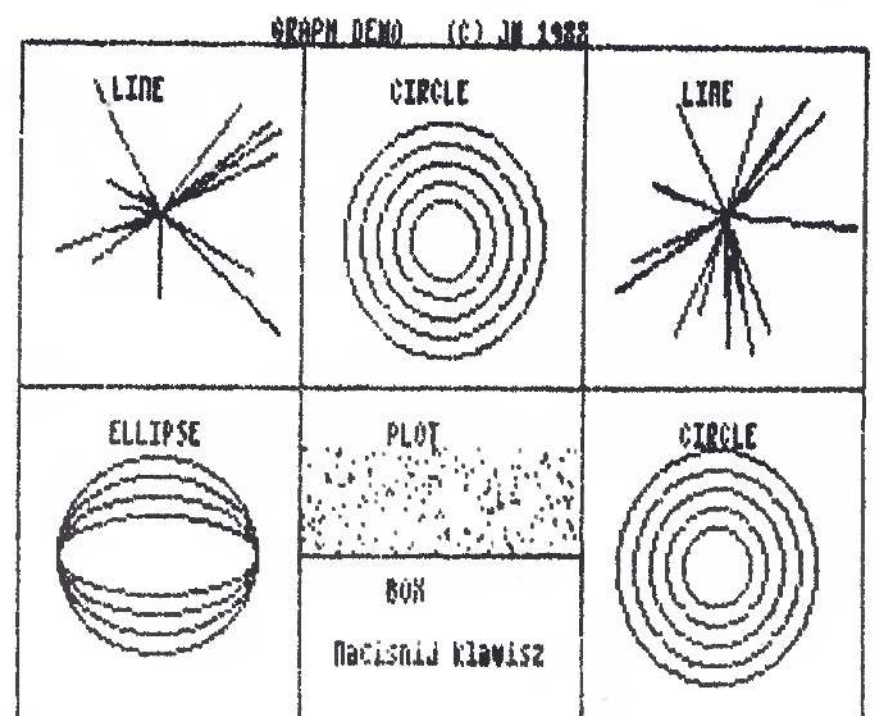
  for i:=1 to 5 do Circle (128,125,5+i*5,1);
  for i:=1 to 4 do Ellipse (42,44,30,5+i*5,1);
  for i:=1 to 5 do Circle (211,41,5+i*5,1);
  for i:=1 to 15
  do Line (42,132,2+Random(80),97+Random(70),1);
  for i:=1 to 200
  do Plot (86+Random(83),45+Random(27),1);
  for i:=1 to 20
  do Line (213,132,174+Random(80),97+Random(70),1);

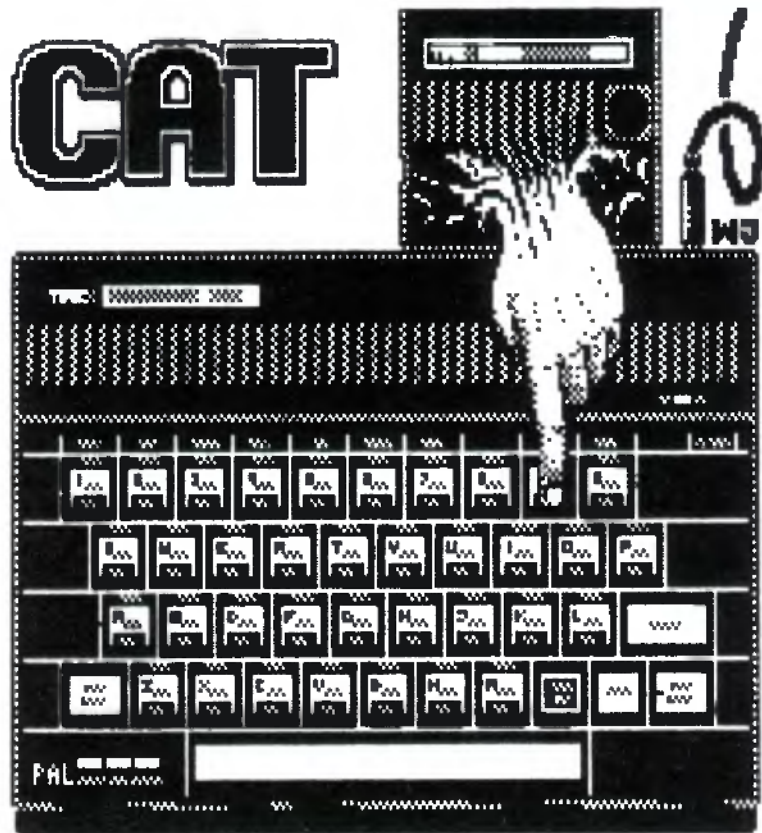
  GotoXY (25,21); write('Nacisnij klawisz ');
  repeat until KeyPressed;
  read(kbd,ch);
  clrscr;
end.

(*****)

```

LISTING 2. Plik DEMO.PAS





**Jest to najlepszy program użytkowy dla TOS'u**

*Brómba*

Niemal każdy system operacyjny, czy to będzie TOS (Timex Operating System) czy też PC-DOS, ma tę wadę, że w celu sprawnej obsługi wymaga od użytkownika sporych zdolności manualnych i dużej pamięci.

TOS, dzięki karkołomnym kombinacjom trybu EXTENDED, klawiszy SHIFT oraz nieodłącznej gwiazdki, stał się szczególnie uciążliwy dla osób, którym zależy na sprawności i szybkości jego obsługi. Dla nich tworzone są programy zwane nakładkami (ang. overlays), które umożliwiają przeglądanie katalogu dysku, uruchamianie wskazanego programu, zmiany jego nazwy lub usunięcie, płynne poruszanie się po podkatalogach, a często dużo więcej. Poniższy program jest próbą realizacji tej idei dla systemu TOS. Jest on napisany całkowicie w kodzie maszynowym, ponieważ musi być krótki, a jednocześnie spełniać jak najwięcej funkcji. Łatwo uruchamia się z poziomu Basica, jak również z assemblera. Jest dobrze związany z systemem, czyli pozostaje aktywnym po instrukcji NEW.

Pierwszym krokiem do otrzymania nakładki jest wpisanie zamieszczonego obok programu do komputera i uruchomienie go. Tworzy on na przygotowanej wcześniej dyskietce plik o nazwie START, do którego kolejno wpisze bajty z linii DATA. Następnie plik jest zamykany i mamy już gotowy program uruchamiany instrukcją **LOAD\* "START"** lub poprzez naciśnięcie klawisza **RESET**.

**DLACZEGO "START"?** Jak widać, procedura jest dosyć skomplikowana. Wpisujemy do komputera program, który tworzy na dyskietce plik o nazwie START, także typu BASIC. Ten zaś, po wgraniu i uruchomieniu się, instaluje dopiero zasadniczy kod maszynowy. Taka konstrukcja umożliwia zastosowanie tej nakładki jako programu uruchamiającego. Jeśli np. mamy na dyskietce pięć gier, to wystarczy ukryć przed wyświetlaniem (za pomocą ATTR\*"nazwa"!) wszystko poza loaderami oraz dołączyć do tego utworzony przez nas program START. Po restarcie systemu na ekranie pokaże się katalog dyskietki, czyli nazwy owych pięciu gier, z których, za pomocą kursorów (↓, ↑), wybieramy jedną i uruchamiamy (ENTER). Osoby używające komputera wyłącznie do gier wiedzą już właściwie wszystko co trzeba i mogą zająć się wpisywaniem programu.

**URUCHAMIANIE.** Zainstalowaną za pomocą programu START (lub inaczej — o czym później) nakładkę uruchamiamy z poziomu Basica za pomocą instrukcji **CAT** (bez gwiazdki!). Z poziomu assemblera jest to bardziej kłopotliwe, ponieważ zasadniczy kod maszynowy znajduje się w pamięci RAM

interfejsu. By osiągnąć cel, należy wpisać następującą sekwencję instrukcji:

```
PUSH IY
LD IY,0
RST 8
POP IY
CALL 8552; uruchomienie nakładki
... ; ciąg dalszy twojego programu
```

**OBSŁUGA.** Po uruchomieniu program wyświetla pierwsze osiem linii katalogu dyskietki. Ma on taki sam format jak standardowy katalog wyświetlony po instrukcji **CAT\***. Reszta ekranu jest pusta, gdyż tam właśnie (przysłonięty atrybutami) jest zapamiętany cały opis katalogu. Taka struktura ekranu jest wynikiem przyjętej konwencji, by nakładka nie ingerowała w pamięć RAM (poza właśnie ekranem), co zapewnia pełną zgodność z prawie całym oprogramowaniem narzędziowym. Katalog przeglądamy za pomocą klawiszy ↑ i ↓. Po katalogu przemieszcza się jednocześnie pasek podświetlający bieżącą nazwę. Nakładka korzysta z tych samych procedur co interpreter Basica, więc każde jej polecenie zastępuje ciąg instrukcji **TOS**. Dostępne funkcje:

KLAWISZ	KOMENDY TOS
[ENTER]	LOAD*"nazwa"
[→]	GOTO*"nazwa": CAT
[←]	GOTO* ↑ ": CAT
[A]	INPUT"NEW""DRIVE\$: GOTO*DRIVE\$d: CAT
[G]	INPUT"GO TO""PATH\$: GOTO*PATH\$: CAT
[E]	ERASE*"nazwa": CAT
[L]	INPUT"LET""NOWA\$: LET*"nazwa" TO NOWA\$: CAT

```
[V] INPUT"V""PATH$:ATTR*PATH$: CAT
[SPACE] CAT
[EDIT] wyjście z programu
[BREAK] powrót do Basic'a z komunikatem
"Nonsense in Basic"
[D] INPUT"DIM""NAME$:
DIM*NAME$: CAT
[P] ATTR*"nazwa" P: CAT
[I] ATTR*"nazwa" I: CAT
[U] ATTR*"nazwa" U: CAT
```

**UWAGA:** W trybie INPUT funkcję klawisza [DELETE] pełni spacja. W przypadku błędu program wyświetla komunikat i czeka na naciśnięcie dowolnego klawisza.

**INSTALACJA.** Są dwa sposoby instalacji. Pierwszy już znamy — uruchomienie programu START. Drugi sposób polega na wgraniu kodu maszynowego nakładki bezpośrednio ze stacji dysków na miejsce, czyli do pamięci RAM interfejsu. Oczywiście w pierw musimy ten kod mieć zapisany na dyskietce. W tym celu po uruchomieniu programu START i powrocie do Basica wpisujemy w trybie bezpośrednim instrukcję **SAVE\*"CAT"CODE 8507,710**. I to już wszystko. Uwaga dla mniej zorientowanych: nazwę bloku "CAT" należy wpisać oczywiście litera po literze, adres zaś 8507 wcale nie jest błędny! Gdy kiedykolwiek będziemy potrzebowali nakładki, a program START zniszczyłby inny program będący właśnie w pamięci, wystarczy wpisać **LOAD\*"CAT"CODE** i wcisnąć [ENTER]. Po wgraniu nakładka nie wymaga żadnej inicjalizacji i jest natychmiast dostępna przy pomocy instrukcji **CAT**.

Wpisanie tego programu to sporo roboty, ale — zapewniam — opłacałnej.

*Wojciech Jabłoński*

```
10 PRINT "Czytanie danych.": LET d$="": FOR n=100 TO 560 STEP 10
20 LET suma=256
30 READ a: IF a<256 THEN LET suma=suma+a: LET d$=d$+CHR$ a: GO TO 30
40 IF suma<>a THEN PRINT "BLAD W LINII ";n: STOP
50 NEXT n
70 DIM *"START": OPEN #1;"START";a:1
80 PRINT "Tworzenie pliku START.": FOR a=1 TO LEN d$: PRINT #1;d$(a):
NEXT a: CLOSE #: CLS : STOP
100 DATA 0,30,0,233,2,233,2,0,0,211,2,249,192,176,34,50,1670
110 DATA 51,55,55,49,34,58,234,253,229,253,33,0,0,207,253,225,2245
120 DATA 33,247,92,17,86,33,237,83,59,33,1,170,2,237,176,195,1957
130 DATA 3,6,0,253,203,1,126,202,130,0,42,93,92,43,126,254,1830
140 DATA 207,194,130,0,225,6,91,205,239,35,62,254,205,29,3,1,2142
150 DATA 22,62,11,17,1,72,213,33,103,33,229,33,134,33,229,205,1680
160 DATA 165,35,0,58,0,33,225,209,254,126,40,11,175,50,100,32,1771
170 DATA 205,197,33,62,145,24,223,6,2,205,185,33,198,3,50,79,1906
180 DATA 33,50,80,33,50,81,33,35,35,35,35,35,62,109,190,32,1184
190 DATA 110,205,185,33,24,238,4,72,205,247,33,62,6,190,32,246,2148
200 DATA 120,201,33,0,32,175,14,32,190,40,11,185,56,4,18,19,1386
210 DATA 14,32,237,160,24,242,27,12,26,254,32,40,249,19,62,1,1687
220 DATA 185,48,13,62,6,18,19,62,17,185,48,4,62,6,18,19,1028
230 DATA 175,18,19,201,33,0,72,175,190,35,32,252,13,32,249,201,1953
240 DATA 175,219,254,230,31,254,31,32,247,205,21,0,40,251,201,58,2505
250 DATA 79,33,79,205,247,33,201,58,79,33,42,80,33,189,201,205,2053
260 DATA 249,35,58,79,33,214,5,205,21,34,205,247,3,35,6,7,1692
270 DATA 197,253,203,87,158,62,3,184,32,11,229,205,26,34,225,56,2221
280 DATA 4,253,203,87,222,205,216,3,35,193,16,228,205,12,34,33,2205
290 DATA 34,34,229,33,103,33,254,11,32,11,205,26,34,254,6,200,1755
300 DATA 61,50,79,33,201,254,10,32,8,205,26,34,60,188,208,24,1729
310 DATA 240,229,254,65,32,6,205,72,35,230,7,1,254,68,32,6,1992
320 DATA 205,72,35,233,2,0,254,71,32,6,205,72,35,236,7,0,1721
330 DATA 254,8,32,13,33,94,0,34,0,32,46,2,205,157,35,7,1208
340 DATA 0,254,86,32,6,205,72,35,86,20,2,254,32,200,254,7,1801
350 DATA 225,193,202,3,6,197,229,87,205,26,34,56,77,188,40,74,2098
360 DATA 122,254,13,32,25,205,211,35,62,12,50,0,33,175,50,2,1537
370 DATA 33,50,77,33,205,193,12,33,180,34,227,205,184,35,254,9,2020
380 DATA 32,5,205,67,35,7,0,254,80,32,5,205,67,35,20,1,1306
390 DATA 254,85,32,5,205,67,35,20,0,254,73,32,5,205,67,35,1630
400 DATA 20,3,254,69,40,9,254,76,40,5,225,225,195,79,34,245,2029
410 DATA 205,237,35,205,18,34,205,247,3,241,254,69,32,25,62,210,2338
420 DATA 205,51,0,175,205,51,0,205,3,34,254,89,194,209,35,205,2171
430 DATA 51,0,205,67,35,4,1,205,211,35,205,78,35,241,5,0,1634
440 DATA 205,211,35,24,85,205,237,35,33,0,32,227,126,35,227,229,2202
450 DATA 205,51,0,205,49,0,225,14,1,229,62,143,119,35,62,32,1688
460 DATA 119,35,175,119,43,43,227,229,197,1,33,22,237,67,136,92,2031
470 DATA 205,247,3,205,3,34,230,127,193,225,227,254,13,40,22,43,2327
480 DATA 13,40,6,254,32,40,211,56,8,12,35,119,121,254,96,48,1601
490 DATA 201,35,12,24,197,175,119,227,225,35,125,205,149,3,227,126,2341
500 DATA 35,227,50,0,33,175,50,2,33,227,126,35,227,205,100,3,1784
510 DATA 205,172,3,48,251,58,2,33,183,225,200,33,33,19,34,136,1891
520 DATA 92,175,50,44,33,33,13,33,205,247,3,205,3,34,225,201,1852
530 DATA 205,18,34,17,0,32,205,200,33,33,9,32,126,254,32,40,1526
540 DATA 3,43,54,46,175,46,12,119,35,201,6,72,33,0,64,54,1219
550 DATA 0,35,124,184,32,249,33,33,24,34,136,92,201,13,0,10,1456
560 DATA 2,0,207,13,0,20,2,0,226,13,0,30,2,0,247,13,1031
```

# JĘZYK MASZYNOWY

## CZ. III.

Omówiona wcześniej procedura „weryfikator” wykorzystuje do policzenia linii programu jedną z ciekawszych procedur ROM-u — NEXT ONE. Rezyduje ona pod adresem 6584 (h19B8).

Na bazie adresu początku programu, określonej linii bądź też początku pola zmiennych (HL) procedura oblicza długość linii lub zmiennej oraz adres następnej linii lub zmiennej. Długość umieszczana jest w rejestrze BC, adres następnej linii/zmiennej w DE.

Ponieważ Basic rozróżnia sześć rodzajów zmiennych, stąd procedura musi rozpoznać siedem rodzajów danych (+ linia) i tak ukierunkować działanie rozkazów, aby obliczyć właściwe parametry. Dlatego zanim rozpoczniemy analizę procedury NEXT ONE, należy dokładnie poznać wszystkie rodzaje zmiennych oraz zasady budowy linii Basica.

Cechy linii programu i zmiennych zawarte są w pierwszym bajcie opisu, którego adres musi znajdować się przed wywołaniem procedury w rejestrze HL.

Każda linia posiada swój numer, zapisany w dwóch pierwszych bajtach opisu linii. Jest on jednak zapisany w niekonwencjonalny sposób, tzn. w kolejności bajt starszy — bajt młodszy, odwrotnie niż jest to przyjęte. Nie jest to pomyłka ani przypadek, lecz celowe działanie. Jeśli bowiem największym numerem linii może być 9999, to starszy bajt linii może przyjąć wartość 0—39, młodszy 0—255 38 razy i 0—15 39 raz. Młodszy bajt nie może więc być wskaźnikiem początku linii. Natomiast śmiało może być nim bajt starszy. I tak, pierwszy bajt każdego z sześciu typów zmiennych jest większy od 39. Dokładnie, wartości tych bajtów rozpoczynają się od 65, a kończą na 250, z tym że pomiędzy nimi istnieją sześciocyfrowe przerwy wynikające z tego, że do oznaczania zmiennych używa się 26 istniejących liter.

### Zmienne

Typy zmiennych oznaczone są cyframi od 2 do 7. Dlaczego nie od 1, wyjaśnimy później.

Numer	Typ
2	zmienna prosta tekstowa; a\$ = "c"
3	zmienna prosta numeryczna jednoznakowa; a = 10
4	tablica numeryczna; a(2)
5	zmienna prosta numeryczna wieloznakowa; xyz = 100
6	tablica tekstowa; a\$(5,10)
7	zmienna sterująca pętlą FOR-NEXT; FOR n=1 TO 15

We wszystkich przypadkach informacja o typie znajduje się w pierwszym bajcie opisu i zajmuje pozycje bitu 5, 6 i 7. Pozostałe pięć bajtów zajmuje numer (nie kod) litery, użytej jako pierwszy znak nazwy zmiennej (koniecznie litera). Jest ich 26, więc i tyle jest numerów; od 1 ("a") do 26 ("z").

26 binarnie to 00011010, więc do zapisu numeru pierwszej litery wystarczy 5 bitów, zapisanych w pierwszym bajcie opisu na pierwszych 5 pozycjach.

Do oznaczania zmiennych stosuje się wyłącznie litery małe, system jednak akceptuje i litery duże, traktując je jako małe.

Tak złożone 8 bitów dają konkretną liczbę z przedziału 65—250. Liczbę dolną otrzymamy dla zmiennej typu 2 (binarnie 010, ale na pozycjach 7, 6 i 5 da 64) oznaczonej literą "a" (numer 1, binarnie 00001 na pozycjach 4, 3, 2, 1, i 0). Analogicznie, liczbę 250 otrzymamy dla zmiennej typu 7 oznaczonej literą "z".

Zauważmy, że gdyby typ pierwszej zmiennej został oznaczony liczbą nie 2, lecz 1, wówczas bit znaczący typu znalazłby się na pozycji 5, której waga wynosi 32. Przy oznaczeniu zmiennej literą "a" (numer 1) otrzymalibyśmy kod równy 33, leżący w zakresie starszych bajtów dopuszczalnych numerów linii.

### Zapis linii w pamięci

Sposób zapisu linii programu w Basic-u w pamięci jest ustalony dla danego typu komputera. W przypadku Spectrum czuwa nad tym interpreter i wyklucza zapis w pamięci linii o składni niezgodnej z przyjętą.

Każda linia posiada numer z zakresu 0—9999, zapisany w dwóch bajtach o przestawionej kolejności. Następne dwa bajty zawierają długość linii i są w normalnym porządku. Długość ta jest liczbą określającą ilość bajtów tworzących opis linii; począwszy od piątego bajtu opisu (za długością), a skończywszy na znaczniku końca (13 — ENTER). Pozostałe bajty opisu zawierają kody słów kluczowych, znaków i liczb zapisane.

Początek programu, tzn. adres młodszy bajtu adresu pierwszej linii, przechowuje zmienna systemowa PROG (23635). Adres końca programu nie jest nigdzie zapisany, można jednak obliczyć go korzystając z faktu, że zmienna VARS notuje adres początku pola zmiennych. Pole to przylega do programu za znacznikiem końca ostatniej linii, więc odejmując od VARS jeden oraz PROG, otrzymamy długość programu.

Dokładny opis składni przykładowej linii podany jest w zeszycie nr 5 Konkretu, był podawany też w „Bajtku”. My zajmiemy się w następnym odcinku zapisem zmiennych w pamięci.

Piotr Sumara

# ROZWIĄZUJEMY K•O•N•K•U•R•S

Po roku, więc z poważnym opóźnieniem, ale z nagrodą, której przez długi czas nie mogliśmy wyegzekwować, rozwiązujemy nasz konkurs świąteczny. Na pomoc Komputerykowi rzuciło się aż... 60 osób. Zwykle dostawaliśmy kilka tysięcy listów, które stały w olbrzymich pudłach na środku redakcji, nie pozwalając na przejście od biurka do biurka. Co się stało tym razem — nie wiemy. Może konkurs był za trudny? Może za mało widoczny?

Za późno już na takie rozważania, przystępujemy więc do rozwiązania. W obecności wysokiej redakcyjnej komisji, w ilości osób dwóch plus czterech obserwatorów, najpierw pytamy autora konkursu, red. Borkowskiego, o tajne hasło, znane tylko jemu. Redaktor Borkowski zastania się kłopotami z pamięcią, ale w końcu stwierdza, że zakodowane hasło brzmiało:

**Na nic Basic i Modula  
Kiedy wiatr po głowie hula.**

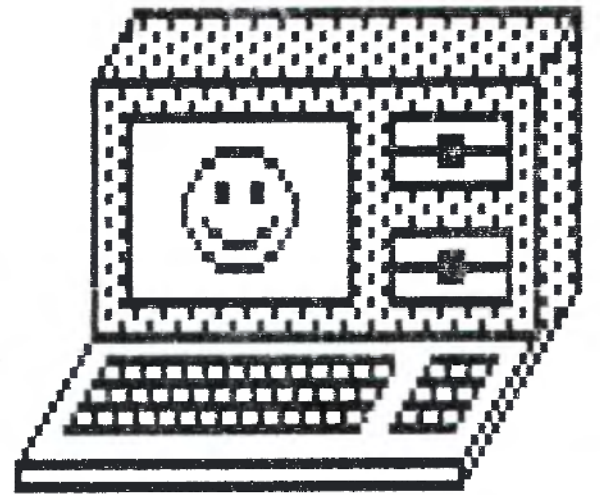
Po oddzieleniu wszystkich pięciu złych odpowiedzi, i zawiązaniu oczu red. Sierotce Przasnyskiej,

następuje oczekiwany od roku moment. Redaktor Przasnyski wygrzebuje ze środka „talii” jedną kartkę pocztową i po rozszlifowaniu szalika odczytuje:

**Robert Biesiada  
ul. Konstytucji 3 Maja 12/30  
28-100 STASZÓW**

Tak więc mamy już zwycięzcę konkursu, któremu serdecznie gratulujemy, a wszystkim 54 pechowcom proponujemy wzięcie udziału w tegorocznym konkursie. Pana Roberta prosimy o kontakt w celu wręczenia nagrody — komputera.

Red.



# SPECTRUMOWA LISTA PRZEBOJÓW

**Oto ostatnie wydanie SLP. Czy zainteresowanie Spectrum spadło? Nie, spadło zainteresowanie naszą Listą. Ostatnie notowanie przedstawia się następująco:**

### Grafika:

1. Hammerfist
2. Purple Saturn Day
3. Savage
4. Netherworld
5. brak

### Muzyka 48:

1. Chase HQ
2. Savage

3. Defender of the Crown
4. brak
5. brak

### Muzyka 128:

1. Tetris 2
2. Altered Beast
3. Ghostbusters
4. Untouchables
5. Thunderbirds

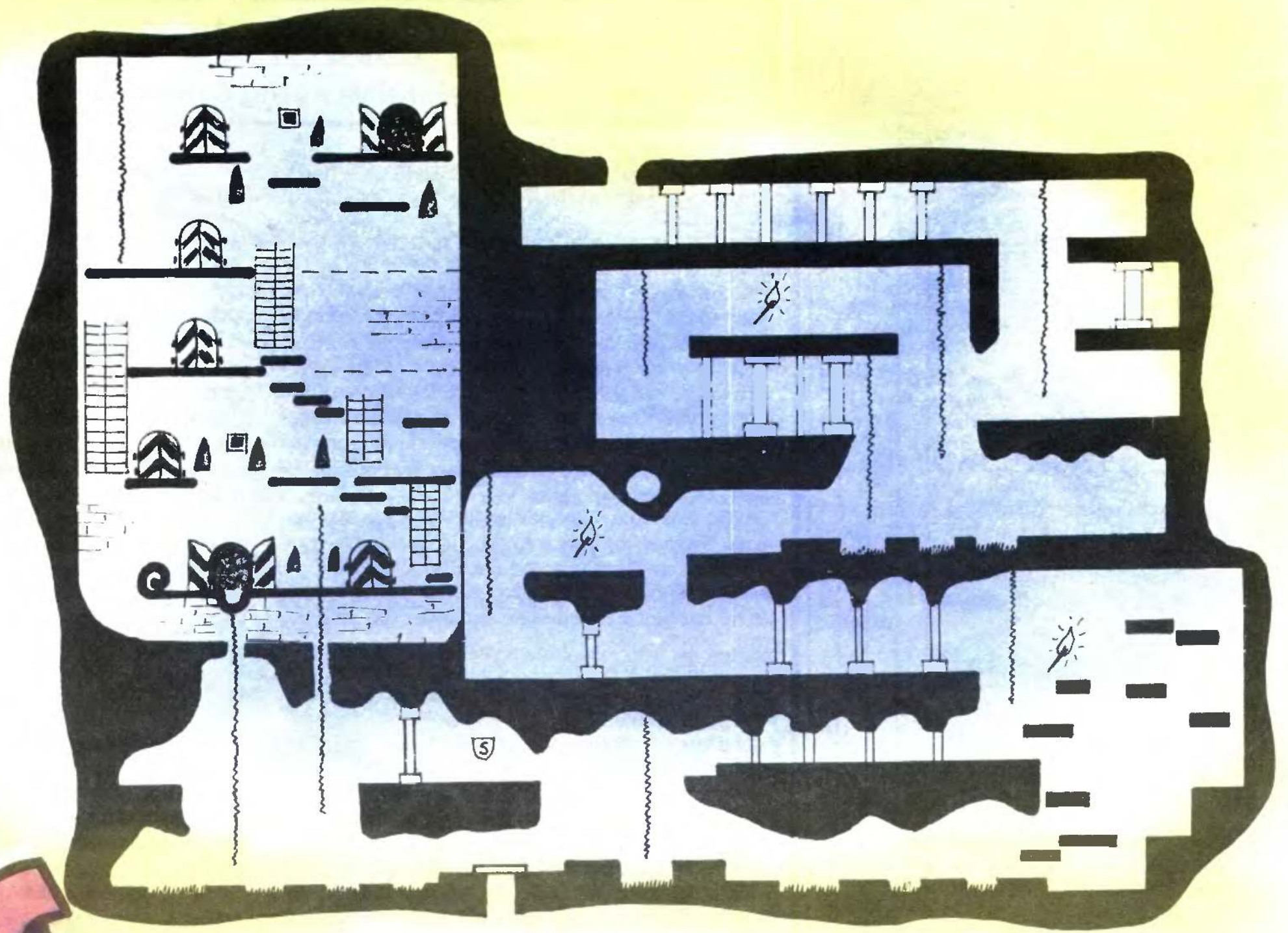
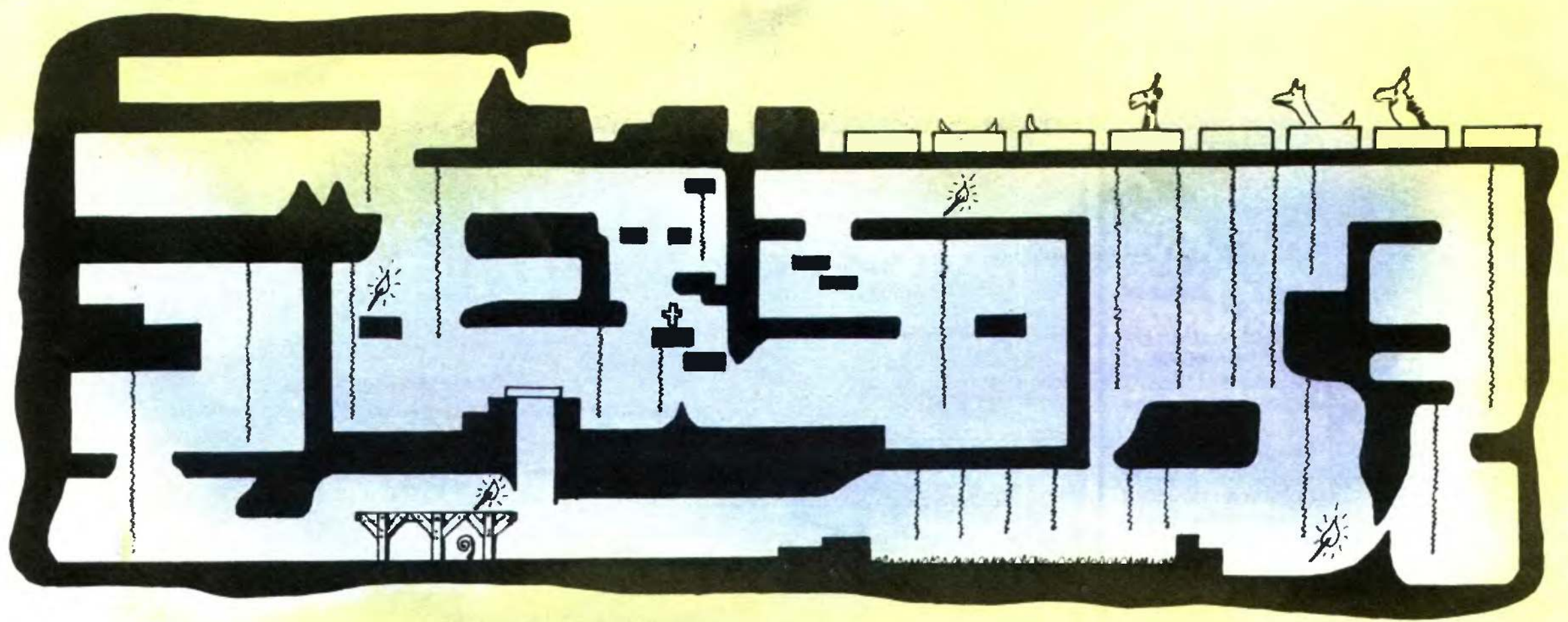
### Użytki:

1. Sound Tracker
2. Text Editor
3. So-So Copy
4. Fox Monitor
5. brak

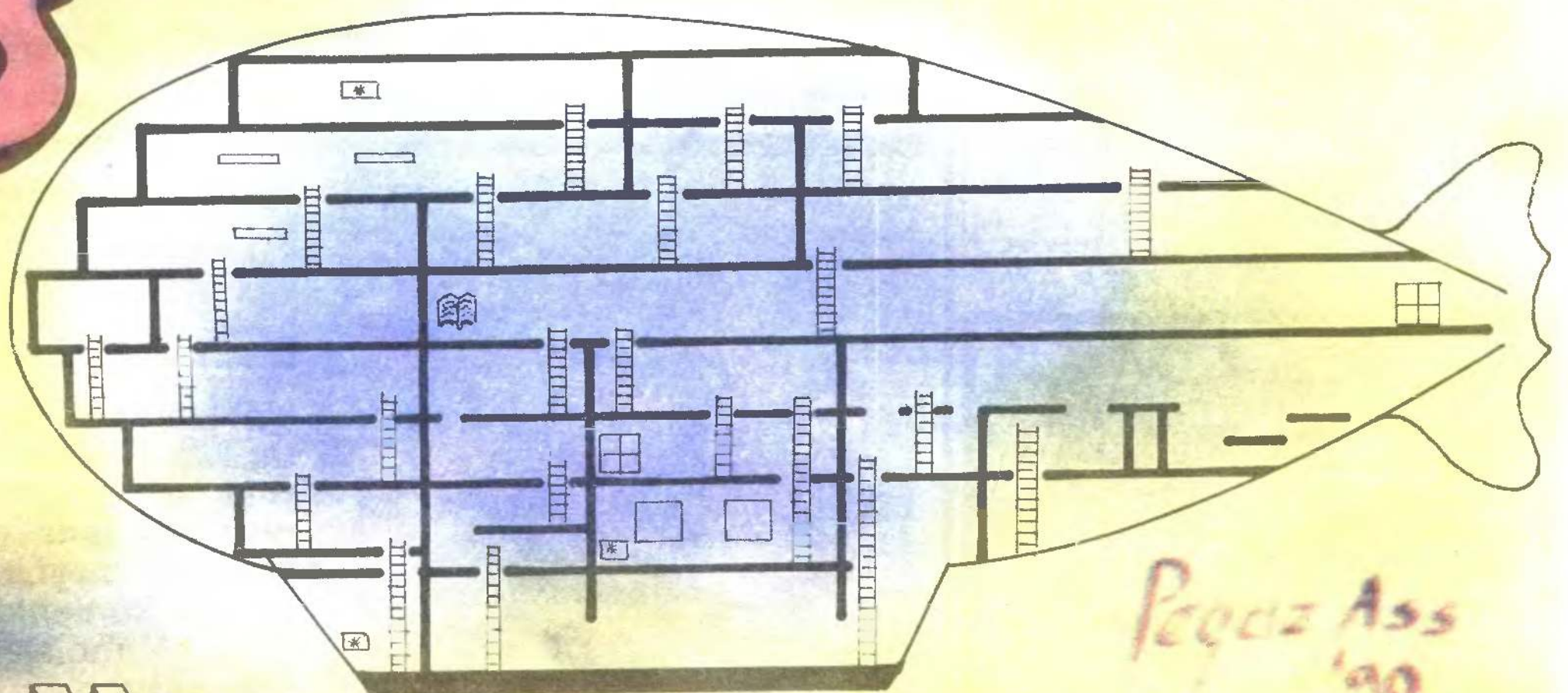


# INDIAN JOE

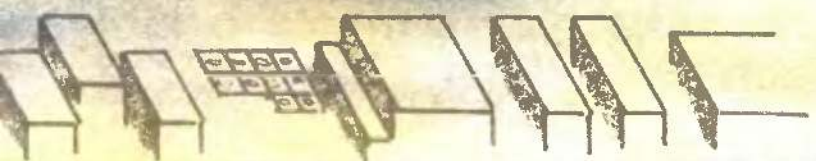




**4**  
**WIVES**



*Peccat Ass '90*



# 10

BAJTKOWA LISTA przebojów 11-12/90

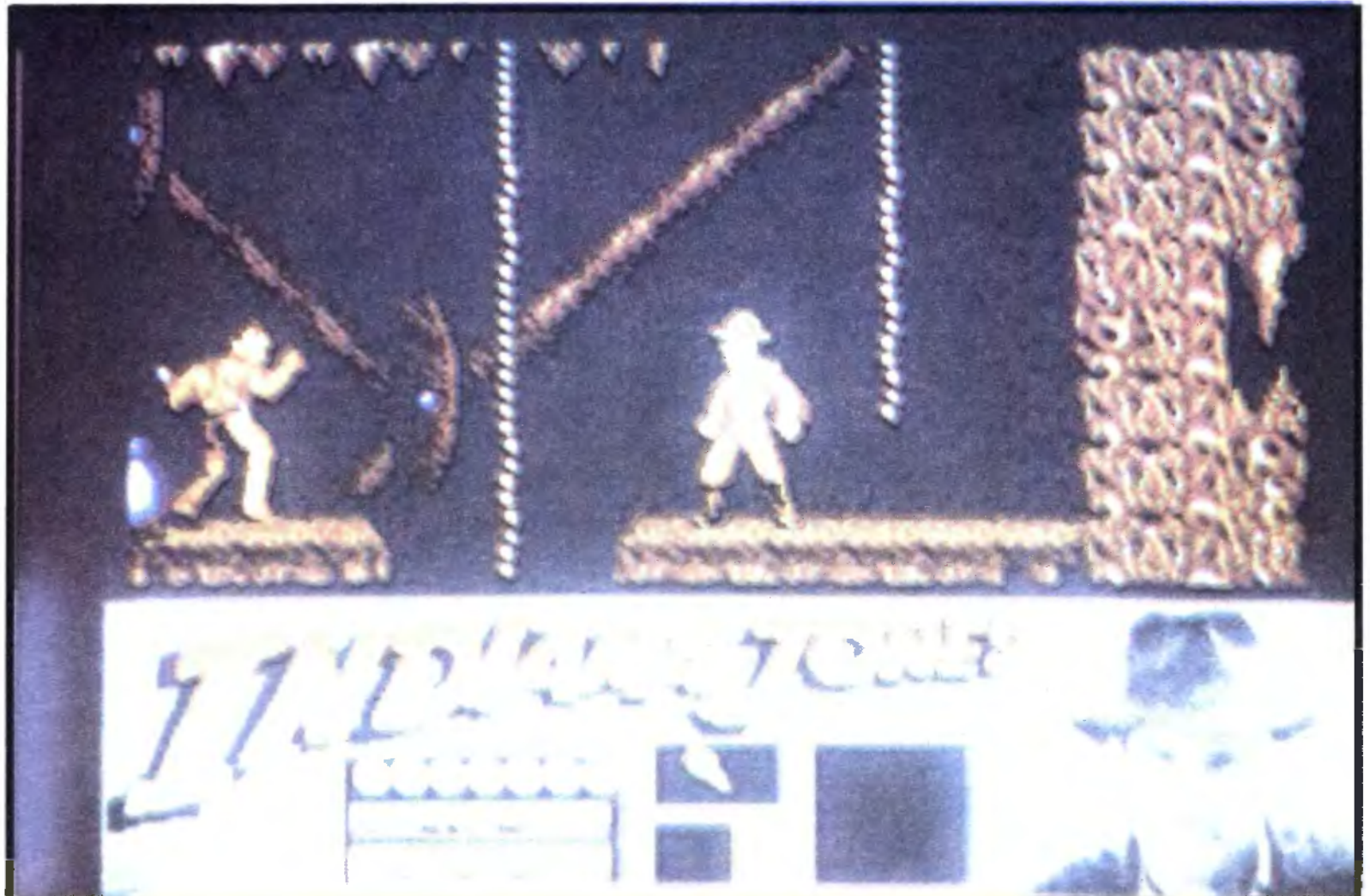
Uwaga, uwaga! To ostatnia Lista Przebojów w „Bajtku”! W numerze styczniowym podsumowanie 1990 roku oraz lat 1986—90, lecz już BEZ comiesięcznego notowania.

Lista zostaje w bardzo rozszerzonej formie przeniesiona do „Top Secret” i odtąd prosimy adresować kartki i listy właśnie na redakcję „Top Secret”. A ostatnie bajtkowe notowanie przedstawia się następująco:

1  
2  
3  
4  
5  
6  
7  
8  
9  
10

- 1 Tetris 2
- 2 Iron Lord
- 3 Untouchables
- 4 Misja
- 5 Hammerfist
- 6 Double Dragon 2
- 7 Batman
- 8 Aliens
- 9 Fighter Bomber
- 10 Cabal

ATARI	AMSTRAD	COMMODORE	SPECTRUM
			x
>	<	x	<
>	<	x	x
<			
>		x	<
>	<	x	<
>		x	x
		x	x
>		x	<
>		x	<



## INDIANA JONES and the Last Crusade

**Przystojny,** rzadko ogolony Harrison Ford to obiekt westchnień wielu pań. Od niedawna, pod postacią gry „Indiana Jones III” również przedmiot pożądań miłośników i miłośniczek gier komputerowych.

Ekranowe przygody Indiany Jonesa fascynowały i wciągały. „Poszukiwacze Zaginionej Arki”, „Świątynia Zagłady” i niedawna „Ostatnia Krucjata” to filmy warte obejrzenia.

Za drugą częścią przybiegła gra, tak kiepska, genialny jest Spielberg. Teraz jednak panowie ze znanej firmy Ocean sprężyli się nieco i na podstawie ostatnich przygód Nieogolonego stworzyli grę, co się zowie!

Akcja postępuje tak, jak w filmie. Na początku młody Indiana zdobywa Krzyż Coronady błędząc po jaskiniach i uciekając po dachu pociągu. Oczywiście wszędobylscy źli wciąż siedzą mu na karku, ale od czego bicz. Jedno smagnięcie i już po gościu.

Po dobrych kilku latach Indiana odszukuje w Wenecji grobowiec Ostatniego Krzyżowca. W wersji na monitorze musi tylko znaleźć wejście opatrzone odpowiednią inskrypcją, porównując ją z datą. Tak czy inaczej, jest to do przejścia.

Potem szybka teleportacja do zamku Brunwald, skąd trzeba wydostać się zabierając bardzo błyszczące COŚ. Tu bardzo przydatny będzie bicz, za pomocą którego można bujać się na kotku.

Teraz Indiana wznosi się sterowcem ponad chmury i mknie do samego Berlina. Niezastąpiony bicz pomoże usunąć z drogi uśmiechniętych hitlerowców i po znalezieniu czegoś BARDZO etap kończy się; wystarczy przejść w prawy dolny róg sterowca.

I znów jak za dotknięciem różdżki Wielki Nieogolony znajduje się zupełnie gdzie indziej, a mianowicie na ścieżce zdrowia prowadzącej do jaskini ze Świętym Graalem. Jedyne przeszkody to dziury w ziemi widoczne i dziury w ziemi niewidoczne, przykryte płytkami. Idąc po płytkach należy kierować się kolejnością liter jak w imieniu IAHWE i zdążyć na czas do Graala. Niestety, nie widać już, czy Indiana pije, czy też nie pije wody z tego Świętego Kielicha, za to koniec gry, fanfary, GAME OVER itp.

Gra jest prosta, jak widać, ale bardzo trudna. Na każdym levelu obowiązuje tzw. limit pochodni, tzn. czas do wypalenia się pokazanej u dołu ekranu pochodni. Otoczenie ciemnieje, ciemnieje i... łup! Jeśli w porę nie znajdziesz nowej pochodni, a nie zdążysz do końca etapu, tracisz życie.

Tych, którzy chcieliby być wspaniali i korzystać z POKE-ów, CHEAT-ów i TRAINER-ów informuję, że jeśli przejdą ostatni etap nie posiadając w sumie czterech przedmiotów (każdy na innym levelu), mogą zapomnieć o sukcesie. Bardziej dociekliwi znajdą jednak rozwiązanie w postaci lepszych POKE-ów, CHEAT-ów itp.

Zainteresowanych mniej zręcznościowymi przygodami Indiany odsyłam do gry Indiana Jones III Adventure, lecz tylko na Amidze i IBM-ie. Gra ta podobna jest w konwencji do Maniac Mansion, zresztą też jest firmy LucasFilm. A jeszcze bardziej zainteresowanych odsyłam do numerów pisma „Top Secret”, gdzie wkrótce ukaże się ta gra — rozpracowana.

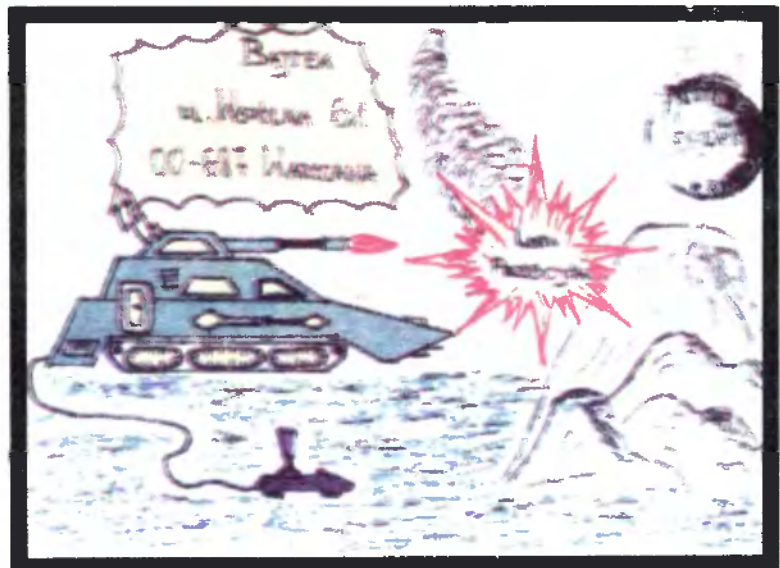
Firma: Ocean  
Komputer: Spectrum, Commodore, Amstrad, Amiga, Atari ST

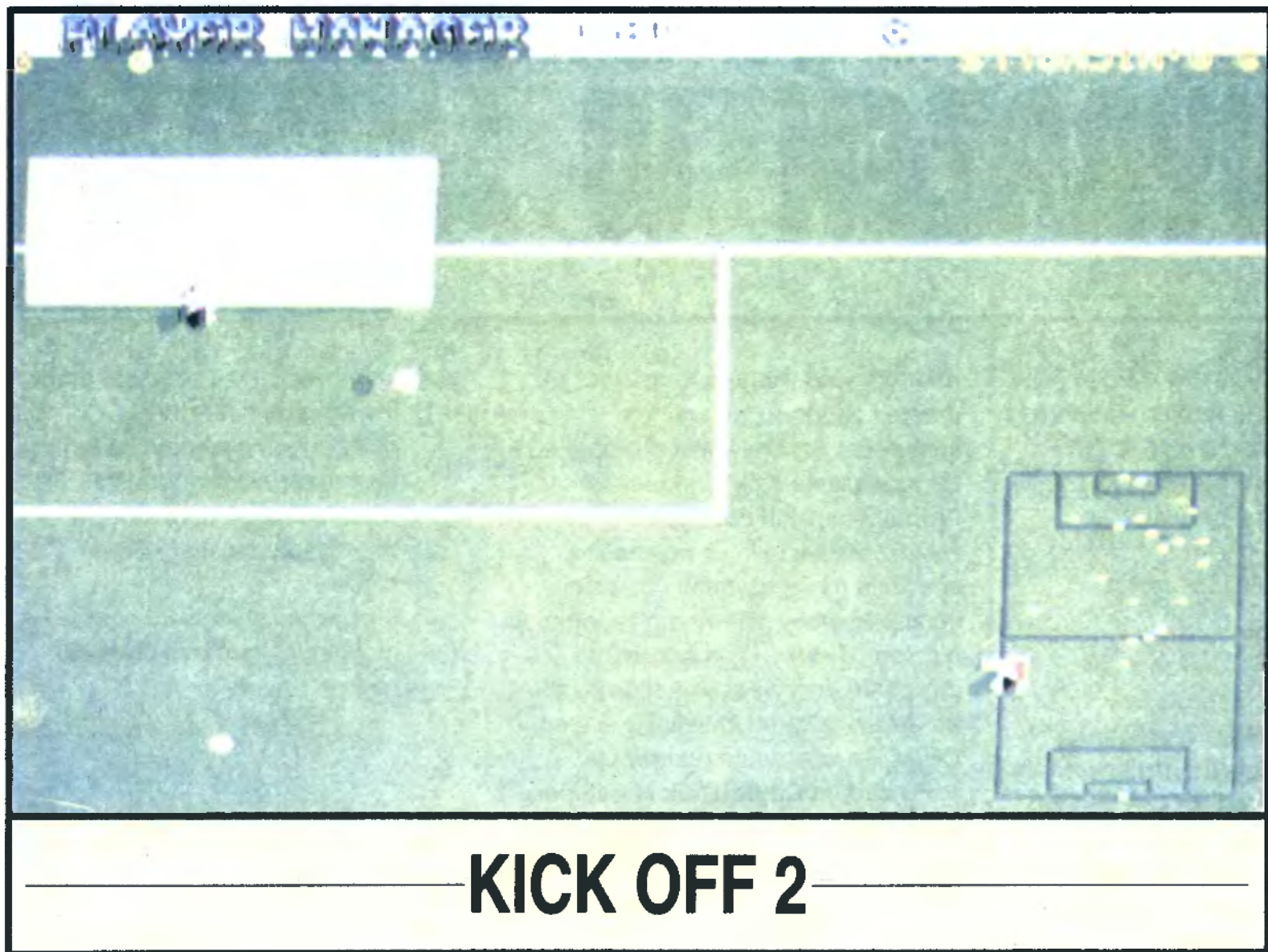
Beata



## PIŁKA W GRZE

Piłkarskie Mistrzostwa Świata, które odbywały się w 1990 roku we Włoszech, były również nie lada gratką dla firm software'owych. Wyczuwając dobrą koniunkturę spowodowaną Mundialem, opracowały one kilka gier z myślą o kibicach piłkarskich.





## KICK OFF 2

**Gwizdek** sędziego. Allen wypuszcza Barkera. Barker podaje piłkę do Quinna. Quinn tańczy z futbolówką na środku boiska. Szybki przerzut do wybiegającego na pozycję Taylora... Strzał! 1:0 dla Anglii!!!

Teraz drużyna przeciwna rzuca się do walki, aby odrobić straty. Przeszkadzają jej w tym znakomici obrońcy, ale końcowy wynik jest uzależniony tylko od Ciebie. Musisz znaleźć właściwą taktykę, wybrać odpowiednie ustawienie i najlepszych graczy. Przed meczem możesz zaaplikować swojej drużynie krótki trening, w ten sposób łatwiej ocenisz możliwości zespołu. Trenuj strzały z przewrotki i stopowanie piłki. Jest o co walczyć. Stawką jest przecież tytuł Mistrza Świata.

Wybierz kolor kostiumów, ustal warunki gry. Możesz zrobić wiele, ale tylko dobra gra zapewni Ci zwycięstwo.

Pamiętaj, by grać fair. Atakuj tylko zawodników z piłką. Nieczysty wślizg na polu karnym kończy się zazwyczaj utratą bramki. Bardzo trudno jest przecieć obronę jedenastki. Szczególnie brutalny faul bywa nagradzany żółtą kartką. Sfaulowany zawodnik może zostać wymieniony przez rezerwowego. W przypadku dużej liczby fauli sędziego ma prawo doliczyć kilkanaście sekund dodatkowo. Mimo że masz naprawdę świetnego bramkarza, to nie dopuszczaj napastników przeciwnika do pozycji strzeleckich. To niebezpieczne.

Po spotkaniu możesz obejrzeć najpiękniejsze strzały lub też cały mecz. To pozwoli Ci uniknąć błędów w następnych spotkaniach. Świetną formą treningu są spotkania towarzyskie. Wybierz odpowiedniego sparring-partnera i ćwicz. Naucz się pro-

wadzić piłkę przy nodze. Wbrew pozorom to bardzo trudna sztuka. Tylko celne podania otworzą Ci drogę do bramki. Przez cały czas możesz kontrolować sytuację na boisku. Skład zespołu ustalasz tuż przed rozpoczęciem meczu. Jest tylko jedna zasada, o której musisz ciągle pamiętać: trening czyni mistrza. Powodzenia.

Opcje:

**PRACTICE** — trening drużyny na boisku z dwoma bramkarzami

**SINGLE GAME** — możliwość wyboru liczby graczy (1-4)

**THE LEAGUE** — spotkania symulacyjne, do użytku tzw. player managera (mogą być wykorzystywane przy wyborze sparring-partnerów)

**THE CUP** — symulacyjne rozgrywki o mistrzostwo świata

**INT. FRIENDLY** — spotkania towarzyskie międzynarodowe

**ACTION REPLAY** — powtórki strzałów lub całego meczu (niezbędny Data Disc)

**KIT DESIGN** — wybór kostiumów piłkarskich oraz narodowości drużyny

**OPTIONS** — wybór warunków gry (rodzaj trawy, siła wiatru, czas gry, tempo gry, ustawienie zawodników w polu, możliwość dogrywki, możliwość wyboru pola, możliwość wyboru poziomu rozgrywek itp.)

**SPECIAL EVENTS** — spotkania specjalne, tzn. o odmiennym charakterze niż Mistrzostwa Świata (niezbędny Data Disc.)

**Firma: Anco**

**Komputer: Spectrum, Commodore, Amiga**

*Shark*

Przeznaczony dla użytkowników ST „Soccer Manager Plus” pozwala sympatykom zmagać na zielonej murawie wcielić się w rolę menadżera drużyny klubowej.

Dźwięczna melodyjka zachęca grającego do podania swojego imienia. Spośród pojawiającego się na ekranie wykazu drużyn wybieramy tę, którą zamierzamy poprowadzić do zwycięstwa. Teraz otrzymujemy kontrakt, który oficjalnie nadaje nam prawa selekcjonera drużyny. Obwarowano w nim wszelkie warunki naszej pracy. Nieudolny menadżer, który narobi zbyt dużych długów lub spowoduje spadek drużyny do niższej ligi, niechybnie już wkrótce będzie wylany z klubu.

Na rozwinięcie działalności otrzymujemy kredyt z banku. Od roztropnego gospodarowania otrzymaną kwotą zależy powodzenie w rozgrywkach. Kapitał można powiększyć sprzedając niektórych zawodników z naszego klubu. Za zdobyte w ten sposób pieniądze zakupimy innych, którzy wzmocnią skład

drużyny. Przed meczem wysyłamy piłkarzy na zgrupowanie treningowe. Najważniejszy moment to ogłoszenie składu jedenastki, która wystąpi w pierwszym spotkaniu. Kierując się skalą punktową określającą wartość poszczególnych zawodników, obsadzamy obronę, atak, linię środkową i ławkę rezerwowych. Wreszcie odbywa się mecz. Komputer pokazuje najciekawsze fragmenty gry i ogłasza rezultat. A potem przygotowania do kolejnego spotkania.

Czasopisma komputerowe, które ukazały się podczas Mistrzostw Świata Italia'90, zachęcały do kupna wielu innych gier. Obok prezentujemy bardzo ładne reklamy opublikowane w 7 numerze miesięcznika „64 er”, przeznaczonego dla fanów Commodore 64. Dotyczyły one gier „World Cup Soccer Italia'90” dla Amigi, ST, C 64 i IBM PC oraz „World Cup 90” dla C 64, Amigi i IBM PC

(JJ)

# S.O.S.

Jestem posiadaczem Commodore 64. Mam problem... Gdy wgram ART STUDIO 2, w żaden sposób nie mogę wgrać rysunku zrobionego na nim. W zamian za informację oferuję wiele gier. **Robert Błażej**, ul. **Derdowskiego 36/4, 71-087 Szczecin**

Mam Atari 800 XL. Poszukuję gry BARBARIAN I i II. W zamian wiadomości do gier: DRACONUS, ZYBEX I i II oraz różne gry. **Andrzej Weiss**, ul. **Cisowa 5, 19-400 Olesko**

Poszukuję gier: BARBARIAN, PLATOON lub AIR WOLF i HENRY'S HOUSE. Cena lub wymiana do ustalenia. **Krzysztof Plewiński**, ul. **II MPS 1/25, 81-661 Gdynia**

Od niedawna mam ZX Spectrum. Poszukuję gier: ROBOCOP, EXOLON, BRUCE LEE. W zamian DE-CATHLON 2, BMX SIMULATOR. **Piotr Górak**, ul. **1 Maja 5/14, 37-100 Łańcut**

Pilnie poszukuję opisów gier UNIVERSAL HERO, ZORRO, SUPER HUEY, SLOT i TWILIGHT WORLD. Proszę też o informację, jak uruchomić grę GHOSTBUSTERS i jak przejść trzecią planszę w MINER 2049. Jestem posiadaczem Atari 65XE. **Piotr Korniejcki**, ul. **Swierczewskiego 10 m 4, 58-500 Jelenia Góra**

W zamian za kasetową wersję GUN FRIGHT, PYJAMARAMA i ROBOCOP na Commodore 64 oferuję BOULDER DASH I i III, PITSTOP I i II, GREEN BERET, THEATRE EUROPE i JET SET WILLY. Wszystkie z nieśmiertelnościami. Bardzo zależy mi na tych grach. **Patryk Modrak**, ul. **Wojska Polskiego 21 m 43, 88-100 Inowrocław**

Jak przejść wąskie przejście w grze POWER BALL i jak grać w TORPEDO RUN? Mam Commodore 16. W zamian służę opisami do HARBOUR ATTACK i FIRE ANT. **Jacek Fidol**, ul. **Spółdzielcza 5, 48-250 Głogówek woj. opolskie**

Poszukuję opisów do następujących gier na Atari 65XE: BLUE THUNDER, FIGHTER PILOT, SPITFIRE 40 i F-15 STRIKE EAGLE. W zamian oferuję 76 opisów do innych gier. **Przemysław Gogolewski**, ul. **Żwirki i Wigury 36, 18-100 Łapy, woj. białostockie**

W zamian za opisy kodów do KENNEDY APPROACH i opisy do SILENT SERVICE i SPITFIRE 40 oferuję opisy do: SOLO FLIGHT, SUPER HUEY, SUBMARINE COMMANDER, TRAP DOOR, STRIKE FORCE COBRA i wielu innych oraz dużo nieśmiertelności. **Stawomir Wojtaszek**, os. **Piaśtów 32 m 7, 31-624 Kraków**

Który z posiadaczy Atari mógłby przysłać mi dokładny opis gry FEUD? W zamian przysyłę inne opisy. **Stawomir Pągowski**, ul. **Wyzwolenia 35 m 39, 70-531 Szczecin**

Proszę o przysyłanie mi kont w banku do gry GHOSTBUSTERS na Atari w wersji kasetowej. Oferuję około 100 programów oraz gier w wersji normalnej i ATARI SUPER TURBO. **Jakub Łuszczek**, os. **Kazimierzowskie 25 m 36, Kraków — NOWA HUTA**

Jestem posiadaczem Atari 65 XE. Poszukuję gier: BARBARIAN, CONAN, KUNG-FU MASTER i THE TRAIN. Wszystkie w wersji kasetowej. W zamian za te gry oferuję 250 innych i 30 programów użytkowych. Pomóżcie! **Hubert Potepa**, ul. **Stasia i Nel 1 m, 10 21-400 Łuków**

Pilnie poszukuję gier: BARBARIAN I i II, WINTER GAMES, GAULET i KUNG-FU MASTER w wersji kasetowej na Atari 65XE. **Jarosław Misiewicz-Murzyn**, ul. **Leśna 12 m, 24 58-530 Kowary**

Błagam!!! Przyślijcie mi OPERATION WOLF w wersji dyskowej na Schneider'a CPC 464. Po przegraniu gry odeślę ją. **Maciej Trojanowski**, ul. **Damrota 8 m 2, 45-064 Opole**

Mam Atari 800XL i magnetofon. Poszukuję gier ROBOCOP, RAMBO, BARBARIAN, KING'S QUEST, DRACONUS. W zamian 20 innych. Bardzo proszę. **Marcin Chymkowski**, ul. **Węglowa 99 m 5, Tychy-Bieruń Nowy**

# ANALIZA WYRAŻEŃ W TURBO PASCALU

**Przy pisaniu programów bardzo często spotykamy się z koniecznością obliczania wartości funkcji.**

Jeśli jej postać jest znana już na etapie tworzenia programu, to możemy bez przeszkód umieścić ją w tekście programu. Jednak w znakomitej większości programów, jak np. rysowanie wykresu, analiza przebiegu zmienności, obliczanie całek — funkcję trzeba wprowadzić do programu z klawiatury.

W tym miejscu pojawia się problem. Zazwyczaj funkcję można w prosty sposób wczytać jako łańcuch znaków, lecz wyliczenie wartości łańcucha sprawia już kłopoty.

Bezpośrednio takiej operacji można dokonać za pomocą funkcji VAL w Basicu ZX Spectrum. Na innych komputerach i w innych językach funkcje VAL oczywiście istnieją, lecz ich możliwości są znacznie ograniczone. Typowo mogą one przypisać zmiennej programu liczbę zapisaną jako łańcuch znaków i nie rozpoznają w nim zmiennych ani standardowych funkcji arytmetycznych.

Rozwiązaniem tego problemu jest napisanie własnego podprogramu, który będzie obliczał wartość dowolnej funkcji. Jego przykładowym rozwiązaniem w języku Turbo Pascal jest przedstawiona poniżej funkcja **Fvalue**.

Funkcja **Fvalue** jest rozszerzeniem standardowej procedury VAL i umożliwia obliczenie wartości funkcji, zapisanej w zmiennej łańcuchowej **Fun** dla maksymalnie trzech zmiennych. Ponadto rozpoznaje 7 zwykłych funkcji arytmetycznych Turbo Pascala. Dozwolona jest dowolna głębokość zagnieżdżenia nawiasów okrągłych i funkcji. Rozpoznawane są funkcje: SIN, COS, ARCTAN, EXP, LN, ABS, SQRT.

Wyliczona wartość będzie poprawna jedynie wówczas gdy argumenty funkcji standardowych będą umieszczone w nawiasach okrągłych. Dodatkowo nie wolno używać „samotnego” minusa dla zmiany znaku. Zapis **-sin(-x)** jest niepoprawny. Aby ominąć tę niedogodność, należy używać zapisu **-1\*sin(-1\*x)**. Zmiennymi muszą być pojedyncze litery X, Y, Z.

Sposób w jaki funkcja **Fvalue** oblicza wartość wyrażenia odpowiadającego wprowadzonej funkcji, przypomina ręczne rachunki, gdyż zwykle złożone wyrażenia

arytmetyczne obliczamy dzieląc je na fragmenty. W kilku krokach stopniowo staramy się uprościć strukturę, wyliczając zawartość nawiasów, funkcji i następnie składamy uproszczone wyniki pośrednie, aż do otrzymania wyniku końcowego.

Funkcja **Fvalue** działa w bardzo podobny sposób. Analizę składni struktury funkcji realizują trzy funkcje: **Czynnik**, **Składnik** i **Wyrażenie**. Dodatkowo funkcja **Czynnik** wywołuje pomocniczą procedurę **Czytajznak** i funkcję **Liczba**.

Czym jest czynnik, składnik, wyrażenie? Najłatwiej odpowiedzieć na to pytanie na przykładzie. Niech funkcja ma postać: **z=4\*x+2/y**. Wyrażeniem jest zawsze cała funkcja, tj. **4\*x+2/y**. Ponadto wyrażenie składa się z sumy lub różnicy składników, zatem składnikami będą **4\*x** i **2/y**. Składnik zaś to iloczyn lub iloraz czynników. Czynnikiem będą więc **4, x, 2, y**.

W bardziej skomplikowanym przypadku, np. **y=2\*sin(4\*x+2)+1**, wyrażeniem jest oczywiście cała funkcja, a zawartość nawiasu **4\*x+2** jest podwyrażeniem. Podwyrażenie musi zostać obliczone w pierwszej kolejności. Składnikiem będzie po obliczeniu wartości podwyrażenia **2\*sin()**. Czynniki — to oczywiście: **2, sin(), 1**. Przy obliczaniu zawartości nawiasu mamy składnik **4\*x** i czynniki **4, x, 2**.

Nietrudno zauważyć, że funkcję **Wyrażenie** należy wywołać jako pierwszą i ją właśnie woła główna funkcja programu **Fvalue**. Całe wyrażenie jest traktowane przez tę funkcję jako suma lub różnica składników. Podobnie funkcja **Składnik** oblicza wartość składnika jako iloczyn lub iloraz czynników.

Funkcja **Czynnik** wylicza wartość najbardziej elementarnej części wyrażenia. Może nią być liczba, zmienna, funkcja lub podwyrażenie umieszczone w nawiasach. W przypadku liczby i zmiennej jest to trywialne. Wartość czynnika będzie równa po prostu tej liczbie lub wartości zmiennej. Jeśli jednak analizowany fragment stanowi funkcja lub jest to lewy nawias, oznacza to, że może tam się znajdować dowolnie skomplikowane wyrażenie. Utożsamianie czynnika z pierwszym napotkanym symbolem (np. z 4 w drugim przykładzie) będzie błędne. W takich przypadkach należy chwilowo zawiesić dotychczasowe obliczenia i wywołać ponownie funkcję **Wyrażenie**, która obliczy zawartość nawiasu.

To kolejne wywołanie funkcji **Wyrażenie** jest oczywiście

rekurencyjne. Należy zauważyć, jak proste i zgodne z naturalnym sposobem dokonywania obliczeń jest to rozwiązanie dzięki rekurencji. Program napisany bez jej wykorzystania byłby z pewnością znacznie mniej czytelny i bardziej skomplikowany. Nie wolno zapomnieć o członie **(\*\$A-\*)** na początku programu, który jest wskazówką dla kompilatora Turbo Pascala w systemie CP/M, że występuje rekurencja.

Procedura **Czytajznak** udostępnia kolejny znak do analizy przypisując go zmiennej **ZNAK**. Duże i małe litery są utożsamiane, spacje zaś pomijane.

Funkcja **Liczba** interpretuje ciąg znaków jako dowolną liczbę całkowitą lub rzeczywistą. Rozpoznaje także postać wykładniczą zapisu liczby.

Przedstawioną wersję funkcji **Fvalue** można rozbudować, co niewątpliwie uczyni ją bardziej uniwersalną. Ewentualna rozbudowa

może odbywać się na przykład przez:

- 1) Powiększenie liczby dostępnych zmiennych i funkcji;
- 2) Eliminację niedogodności z „samotnym” minusem;
- 3) Wprowadzenie diagnostyki błędów;
- 4) Możliwość stosowania nazw zmiennych o różnej długości.

Robert Magdziak

Literatura:

1. R. Waclawek, Kurs języka Pascal, cz. 16, „Młody Technik” 6/87.
2. N. Wirth, Algorytmy + struktury danych = programy, WNT 1989.

```

(*****)
{
{ Analiza wyrazen arytmetycznych.Turbo Pascal 3.0, CP/M 2.2 }
{
(*****)

{$A-}
program Analiza;

type funkcja=string[255];
var
    znak :      char;
    f,x,y,z:    real;
    i :        byte;
    fun :      funkcja;
    error :    integer;

procedure Czytajznak(fun:funkcja);
(*****)
{ udostepnia kolejny znak do analizy jako zmienna ZNAK }
(*****)
begin
    repeat
        znak:= upcase(fun [i]);
        i:=i+1
    until znak<>' ';
end; { procedury Czytajznak }

function Liczba (fun: funkcja) :real;
(*****)
{ interpretuje ciag znakow jako dowolna liczbe }
(*****)
var
    licz:      real;
    z: string[25];

begin
    z:='';
    while (znak in ['0'..'9','E','e','.',',','-','+']) and
        not((z[length(z)] in ['0'..'9'])and(znak in ['-','+']))
    do
        begin
            z:=z+znak;
            Czytajznak(fun);
        end;
    val(z,licz,error);
    if error>0 then writeln(' zla postac liczby !');
    liczba:=licz
end; { funkcji Liczba }

function Wyrażenie(fun:funkcja) :real; forward;
(*****)
{ wyrażenie jest suma lub roznica skladnikow }
(*****)

```



```

function Czynniki(fun:funkcja):real;
{*****}
{ czynnikiem jest liczba, zmienna lub funkcja }
{*****}
var
  nazwa: string[6];

begin
  Czytajznak(fun); nazwa:='';
  case znak of
    '(' : begin
      Czynniki:=Wyrazenie(fun);
      if znak=')' then Czytajznak(fun)
      else
        begin
          writeln('brak nawiasu !');
          error:=1;
          end;
        end;
    end;

    'X' : begin Czynniki:=x; Czytajznak(fun) end;
    'Y' : begin Czynniki:=y; Czytajznak(fun) end;
    'Z' : begin Czynniki:=z; Czytajznak(fun) end;

    'S','C',
    'A','E',
    'L': begin
      { sa to litery poczatkowe wszystkich funkcji }
      while znak in ['A'..'Z'] do
        begin
          nazwa:=nazwa+znak;
          Czytajznak(fun)
        end;
        if nazwa='SIN' then Czynniki:=sin(Wyrazenie(fun));
        if nazwa='COS' then Czynniki:=cos(Wyrazenie(fun));
        if nazwa='ARCTAN' then Czynniki:=arctan(Wyrazenie(fun));
        if nazwa='EXP' then Czynniki:=exp(Wyrazenie(fun));
        if nazwa='LN' then Czynniki:=ln(Wyrazenie(fun));
        if nazwa='ABS' then Czynniki:=ABS(Wyrazenie(fun));
        if nazwa='SQRT' then Czynniki:=sqrt(Wyrazenie(fun));
        if znak=')' then Czytajznak(fun);
        end;
      else Czynniki:=Liczba(fun);
      end;
    end; { funkcji Czynniki }

function Skladnik(fun:funkcja) :real;
{*****}
{ skladnik jest iloczynem lub ilorazem czynnikow }
{*****}
var s:real;

begin
  s:=Czynniki(fun);
  while znak in ['*','/'] do case znak of
    '*':s:=s*Czynniki(fun);
    '/':s:=s/Czynniki(fun);
    end;
  skladnik:=s
end; { funkcji Skladnik }

function Wyrazenie;
var w:real;

begin
  w:=Skladnik(fun);
  while znak in ['-','+'] do case znak of
    '+':w:=w+Skladnik(fun);
    '-':w:=w-Skladnik(fun);
    end;
  wyrazenie:=w
end; { funkcji Wyrazenie }

function Fvalue(fun:funkcja):real;
{*****}
{ funkcja podstawowa, oblicza wartosc funkcji }
{*****}
begin
  i:=1;
  Fvalue:=Wyrazenie(fun)
end; { funkcji Fvalue }

begin
  fun:=''; error:=0;
  writeln('Podaj Funkcje ?');
  readln(con,fun);
  writeln('Podaj wartosci zmiennych X,Y,Z ?');
  readln(x); readln(y); readln(z);
  f:=Fvalue(fun);
  if error=0 then writeln(' F(X,Y,Z)=',f:10:6)

end. { programu Analiza }

```

# TYLKO

## DLA POCZĄTKUJĄCYCH

Ten artykuł jest tylko i wyłącznie dla początkujących, takich naprawdę „zielonych”. Jeśli już coś wiesz, możesz go spokojnie nie czytać.

Masz komputer Amstrad i nie bardzo wiesz, co z nim zrobić? Hmm... Zaczniemy od przygotowania komputera do pracy. Zestaw składa się z komputera (pudełko z klawiszami) i monitora. Jeśli masz komputer „fabrycznie nowy” to sprawdź, czy sznur sieciowy jest zakończony wtyczką! Anglicy mają zwyczaj dostarczać wszystko bez wtyczek, więc sprawdź. Jeśli nie masz wtyczki, to ją załóż (jeśli nie potrafisz, poproś ojca, starszego brata albo sąsiada).

Krok drugi: z przodu monitora są dwa kabełki. Należy je podłączyć do gniazdek w tylnej ścianie komputera (nie da się pomylić, bez obawy). Komputery CPC 664 i CPC 6128 mają jeszcze trzeci kabełek, wychodzący z tyłu komputera. Trzeba go podłączyć do gniazdka w przedniej ścianie komputera.

Z przodu monitora, poniżej kineskopu jest przycisk z napisem POWER. Wciśnij go — w ten sposób włączysz monitor oraz zasilacz komputera. Teraz znajdź przełącznik POWER na tylnej ścianie komputera i przesuń go. Zapaliła się lampka? Jeśli nie, to wyłącz monitor i sprawdź, czy jest podłączony do prądu! Mogę chyba założyć, że wszystko działa i na ekranie pojawiła się „wizytówka” komputera? Wyreguluj teraz obraz pokrętkami BRIGHTNESS (jasność), CONTRAST (kontrast) i V-HOLD (synchronizacja pionowa) i ruszamy dalej.

Na prawo od klawiatury znajduje się magnetofon lub stacja dysków. Obsługi magnetofonu chyba nie muszą tłumaczyć, stacja wymaga jednak opisu.

Używamy dyskietek 3-calowych,

nie mylić z 3.50-calowymi (Amiga, Atari ST). Dyskietkę należy wsunąć do oporu (słychać odgłos „zatraskiwania się”). Aby wyjąć dyskietkę, trzeba wcisnąć przycisk (na lewo poniżej otworu). Stacja wyposażona jest także w kontrolkę pracy. Gdy zaświeci ona, NIE WOLNO wyjmować dyskietki, bo można uszkodzić głowicę.

Co teraz? Spróbujmy zobaczyć co jest na taśmie lub dyskietce. Piszemy CAT i naciskamy RETURN. Na CPC 464 pojawi się napis „Press PLAY then any key” — trzeba nacisnąć klawisz PLAY magnetofonu a potem dowolny klawisz (ale nie ESC!). Gdy komputer natrafi na program, wyświetli „Found NAZWA block” i numer bloku (programy nagrywane są w blokach po 2 kilobajty). Spróbuj znaleźć pierwszy blok jakiegoś programu i cofnij nieco taśmę. Jeśli nie chcesz dalej katalogować programów, naciśnij ESC.

W przypadku CPC 664 i CPC 6128 po komendzie CAT wyświetlony zostaje katalog dyskietki, zaczynający się od „Drive A: user 0” (dysk A: użytkownik nr 0). Poniżej będzie spis programów wraz z ich długościami, na końcu informacja, ile kilobajtów można jeszcze zapisać na tej dyskietce.

Jeśli masz CPC 464 i ustawiłeś taśmę przed pierwszym blokiem jakiegoś programu, naciśnij jednocześnie CONTROL i „małe” ENTER (w bloku cyfrowym klawiatury). Pojawi się rozkaz „RUN” oraz „Press PLAY...”. Naciśnij dowolny klawisz, a program zacznie się ładować i wystartuje po załadowaniu.

W przypadku CPC 664 i CPC 6128 musisz napisać RUN, cudzysłów i nazwę programu, a następnie nacisnąć RETURN. To tyle na wstępie, więcej znajdziesz w instrukcji (wydano ją także po polsku!).

Michał Szokoło



## PROCEDURY SYSTEMOWE AMSTRADA

Kiedyś, dawno dawno temu (dokładnie w roku 1987), był już taki cykl. Druk listy procedur urwał się w dwóch trzecich. Czytelnicy w listach zwracają uwagę, że tak nie można, trzeba doprowadzić do końca to co się zaczęło. A więc, od tego numeru następuje CIĄG DALSZY. Uwaga: podajemy adresy wspólne dla wszystkich CPC!

Jak korzystać z tych procedur? W przypadku procedur nie wymagających paramet

trów wejściowych, można użyć rozkazu "CALL" w BASIC-u (np. "CALL &BCA7"). Inne procedury można wywoływać tylko na poziomie asemblera. Przykład: aby wydrukować znak (z poziomu asemblera), trzeba wywołać procedurę TXT PRINT CHAR (rozkazem "CALL #BB5A"), umieszczając wcześniej kod znaku w akumulatorze (rozkaz "LD A, znak"). Dalsze przykłady już wkrótce!

### PAKIET DŹWIĘKU (SOUND PACK)

#### SOUND RESET — RESET PAKIETU DŹWIĘKU

#BCA7

Kasuje aktualny dźwięk, kolejki dźwięków i obwiednie.

Wejście: nie ma.

Wyjście: Rejestry AF, BC, DR, HL skasowane.

#### SOUND QUEUE — DŹWIĘK DO KOLEJKI

#BCAA

Dopisuje podany dźwięk do odpowiedniej kolejki.

Wejście: HL zawiera adres bloku dźwięku.

Wyjście: jeśli CY=0 to operacja nieudana, HL niezmienniczone, jeśli CY=1 to operacja udana. Rejestry AF, BC, DE, HL i IX skasowane.

Format bloku dźwięku:

bajt 0: kanał (jak w BASIC-u)  
bajt 1: obwiednia głośności (amplitudy)  
bajt 2: obwiednia tonu  
bajty 3,4: okres dźwięku (12 bitów)  
bajt 5: okres szumu  
bajt 6: początkowa głośność  
bajty 7,8: długość lub ilość powtórzeń  
Zero oznacza wartość domyślną lub pominięcie parametru.

#### SOUND CHECK — SPRAWDŹ STATUS KANAŁU

#BCAD

Odczytuje status kanału i ilość wolnych miejsc w kolejce.

Wejście: A zawiera bit oznaczający kanał (patrz instrukcja).

Wyjście: A zawiera bajt statusu, znaczenie bitów poniżej:

bity 0,1,2: ilość wolnych miejsc w kolejce

bit 3: oczekiwanie na kanał A  
bit 4: oczekiwanie na kanał B  
bit 5: oczekiwanie na kanał C  
bit 6: dźwięk w kanale jest zatrzymany  
bit 7: aktywność kanału (czy gra w danej chwili)  
Rejestry BC, DE, HL i flagi skasowane.

#### SOUND ARM EVENT — WŁĄCZ PRZERWANIE DŹWIĘKOWE

#BCB0

Aktywuje zdarzenie, które wywoła procedurę użytkownika, jeśli pojawi się wolne miejsce w kolejce.

Wejście: A zawiera bit oznaczający kanał, HL zawiera adres bloku zdarzenia

Wyjście: AF, BC, DE i HL skasowane. Porównaj: KL INIT EVENT

#### SOUND RELEASE — UWOLNIJ DŹWIĘK

#BCB3

Uwalnia dźwięk zablokowany ustawieniem bitu zatrzymania.

Wejście: A zawiera bit/bity określające kanały.  
Wyjście: AF, BC, DE, HL i IX skasowane.

#### SOUND HOLD — ZABLOKUJ WSZYSTKIE KANAŁY

#BCB6

Natychmiast wstrzymuje granie wszystkich dźwięków i blokuje kanały.

Wejście: nie ma.

Wyjście: CY=1 jeśli był aktywny (grany) jakiś dźwięk.  
AF, BC, HL skasowane.

#### SOUND CONTINUE — ZWOLNIJ ZATRZYMANE DŹWIĘKI

#BCB9

Usuwa skutki procedury SOUND HOLD. Wstrzymane dźwięki będą dalej grane.

Wejście: nie ma.  
Wyjście: AF, BC, DE i IX skasowane.

#### SOUND AMPL ENVELOPE — USTAW OBWIEDNIĘ GŁOŚNOŚCI

#BCBC

Definiuje programową obwiednię głośności (amplitudy).

Wejście: A zawiera numer obwiedni. HL wskazuje na blok danych.

Wyjście: CY=1 jeśli wykonane poprawnie,

HL zwiększone o 16. A i BC skasowane.  
Blok danych składa się z bajtu zawierającego ilość sekcji (od 1 do 5) oraz z odpowiedniej liczby 3-bajtowych sekcji. Kolejne bajty sekcji oznaczają: ilość kroków, rozmiar kroku, opóźnienie.

#### SOUND TONE ENVELOPE — USTAW OBWIEDNIĘ TONU

#BCBF

Definiuje obwiednię tonu (częstotliwości). Parametry wejściowe i wyjściowe tak jak

SOUND AMPL ENVELOPE

#### SOUND A ADDRESS — ADRES OBWIEDNI GŁOŚNOŚCI

#BCC2

Odczytuje adres obwiedni głośności (amplitudy).

Wejście: A zawiera numer obwiedni.

Wyjście: jeśli CY=1 to operacja udana, HL

zawiera adres a BC długość obwiedni; CY=0 oznacza błąd. HL skasowane, BC zachowane. Akumulator zawsze skasowany.

#### SOUND T ADDRESS — ADRES OBWIEDNI TONU

#BCC5

Odczytuje adres obwiedni tonu (częstotliwości).

Parametry jak dla SOUND A ADDRESS.

Na tym kończy się pakiet dźwiękowy. W na-

stępnym odcinku pakiet systemu operacyjnego (KERNEL). Na podstawie "The Ins & Outs of the AMSTRAD"

M. Szokoło

# WYDRUK EKRAŃU

## NA CPC 6128

**Pewnego razu ujrzałem takie zdarzenie: facet pracował na Amstradzie PCW 8256. Zrobił wykres sinusa w Turbo Pascalu, potem nacisnął jakieś dwa klawisze i... na drukarce pojawiła się kopia ekranu! Zaczęła zżerać mnie zazdrość. Powiedziałem sobie: MUSISZ MIEĆ COŚ TAKIEGO NA CPC 6128. A potem siadłem i napisałem. Program działa i jest krótki! A więc do dzieła.**

Potrzebne będą: edytor tekstu, asembler M80.COM i linker. Wpisujemy program w edytorze, pod nazwą SCOPY.MAC i nagrywamy go na dysk. Następnie asembujemy go rozkazem M80=SCOPY i otrzymujemy plik SCOPY.REL, zawierający kod w postaci relokowalnej. Następnie linkujemy (konsolidujemy, jeśli kto woli) rozkazem LINK SCOPY[0C NR] i otrzymujemy gotowy program SCOPY.COM (literki w nawiasach kwadratowych to opcje linkera). Jeśli używamy asemblera GEN80, to należy zastąpić dyrektywy ".PHASE" dyrektywami "ORG" i pominąć oba wystąpienia ".DEPHASE" oraz zmienić zapis liczb szesnastkowych. Uruchomienie polega na napisaniu "SCOPY" i naciśnięciu klawisza RETURN. Program ładuje się i wyświetla wizytówkę. Teraz wystarczy włączyć drukarkę, włożyć papier i nacisnąć jednocześnie klawisze CONTROL oraz COPY. Drukarka zacznie reprodukować zawartość ekranu na papierze, a BORDER będzie migał (taki mały bajerek).

Jak działa ten program? Procedura INSTAL umieszcza część relokowalną w banku systemowym i włącza ją do kolejki przerwania. Kiedy wystąpi przerwanie, procedura PRZERWANIE sprawdza, czy naciśnięto klawisze nr 9 i 23 (CONTROL i COPY), robiąc to w sposób nieco trickowy (bezpośrednie badanie mapy bitowej), ale bardzo szybki. Jeśli nie naciśnięto jednocześnie w/w klawiszy, procedura wchodzi do systemu. W przeciwnym wypadku program przechodzi do procedury HARDCOPY.

Tu ustalany jest adres bajtu w lewym dolnym rogu ekranu, od którego rozpoczyna się drukowanie oraz ustawiany licznik pasków do wydrukowania. Paski mają szerokość ośmiu bitów, są to kolumny po 200 bajtów brane pionowo od dołu do góry. Przechodzimy do procedury DRUKUJ, która drukuje znak firmowy i ustawia wysuw papieru na 8 pixeli. Teraz program wchodzi w pętlę, która powtarza się 80 razy, drukując po jednym pasku za każdym przejściem.

Na początku robiony jest pięciodziesięciodziesiąty margines i ustawiane parametry paska: początkowa gęstość, 1600 punktów szerokości. Teraz do HL bierzemy adres bajtu, a do B —

licznik bajtów w pasku (200).  
Powtarzamy B razy: pobierz bajt,  
wyślij go 8 razy, przestaw adres  
na następny (o linię w górę).  
W ten sposób wystaliśmy już cały  
pasek! Należy więc obliczyć  
adres początkowy następnego  
paska — o bajt w prawo od  
poprzedniego, zmniejszyć licznik  
pasków o jeden i sprawdzić, czy  
trzeba drukować pasek następny.  
Jeśli tak, wracamy do etykiety  
DRUKUJ, jeśli zaś nie, drukujemy  
margines (przejdź do nowej  
linii), wysuwamy papier do  
początku nowej strony  
i dzwoniemy na fajrant. Uff, koniec!

Z tym krótkim programikiem być  
nie mały kłopot. Musiał być  
w banku systemowym (bank 0),  
a tam nie ma aż tyle miejsca

(w jednym kawałku). Co trzeba  
zrobić? Wyrzucić pół generatora  
znaków! Znaki o kodach 0—31  
są niewidoczne, a znaki 32—127  
prawie takie, jak znaki  
standardowe. Przełączyłem więc  
pakiet tekstowy tak, że wzorce  
znaków 0—127 bierze z ROM-u  
systemowego, a tylko znaki  
128—255 z generatora znaków  
systemu CP/M Plus. W ten  
sposób dostałem cały kilobajt!  
Niestety, niektóre programy tego  
nie lubią... Na szczęście  
nieliczne, więc SCREEN COPY  
da się zastosować często.

Tak przy okazji, dobrze mieć  
wyjście Centronics przerobione  
na osiem bitów. Wydruki będą  
ładniejsze.

*Michał Szokoło*

```

; Screen Copy -- Drukowanie Ekranu
; Copyright (c)1990 by M. Sz. & "BAJTEK"
; Ten program jest PUBLIC DOMAIN!
;
; .Z80
boot: ld a,(7) : ostatni bajt
sub 0C2H
jr nc,blad ; dość pamięci?
ld hl,segmenty
ld de,instal
ld bc,inst_len+ph_len
ldir : kopiuj do COMMON
call instal : i uruchom
jp 0

blad: ld de,blad_txt
ld c,9 : druk tekstu
jp 5 : wykonaj i wyjdź

blad_txt: db 13,10,"Brak pamieci!$"

segmenty:
instal: .phase 0C000H ; w segmencie COMMON
xor a
call 0FD04H ; włącz bank 0
ld hl,0FF80H
ld (0B734H),hl : kasuj znaki 0-127
ld hl,8400H
ld (0B736H),hl : zostaw 128-255
ld hl,int_block : odłącz blok
call 0BCDDH : przerwania
ld hl,ph_start
ld de,przerwanie
ld bc,ph_len : prześlij kod
ldir : rezydentny
call witaj
ld a,9 : Ctrl-COPY bez genera-
ld b,0FFH : cji znaku
call 0BB33H
ld bc,81FFH
ld hl,int_block
ld de,przerwanie
call 0BCD7H : włącz przerwanie
ld a,1
call 0FD04H : bank 1
ret : powrót

witaj: ld hl,hello : powitanie: drukuj
witaj1: ld a,(hl) : tekst
cp 0FFH
ret z
inc hl
call 0BB5AH : TXT PRINT CHAR
jr witaj1

hello: db "V"-64,1,"Screen Copy "
db "(c)1990 M. SZOKOL".8,"/O "
db '& "BAJTEK" ',"V"-64,0
db 13,10,0FFH

inst_len equ $-instal
ph_start:
.dephase : koniec instalatora

keymap equ 0B63FH : mapa bitowa klawiat.
size equ 200 : długość paska

przerwanie: .phase 8000H : kod rezydentny
ld a,(keymap+1) : przerwanie: czy
cp 0FDH : naciśnięto CTRL i
ret nz : COPY ?
ld a,(keymap+2)

```

```

cp 7FH
ret nz : wróć jeśli nie
di
ld (sav_sp),sp : zapamiętaj stos syst.
ld sp,sav_sp : utwórz własny stos
hardcopy: ld hl,24 : adres pozycji (1.25)
call 0BC1AH
rept 7 : siedem linii nizej
call 0BC26H
endm
ld (ekran),hl : zapamiętaj adr. start
ld a,80 : licznik pasków
ld (licznik),a : zapamiętaj
jp drukuj

sygnal: ld bc,7F10H : migotanie BORDERU
out (c),c : sygnalizuje pracę
ld a,r
and 31
or 64
out (c),a

print: push af : drukuj znak
push hl
push bc
ld b,0F5H : drukarka gotowa?
in c,(c) : czekaj na gotowość
bit 6,c
jr nz,busy?
send: call 0BD31H : wyślij znak
pop bc
pop hl
pop af

setup: ld hl,setup_str : ustaw wysuw papieru
setup1: ld a,(hl)
cp 0FFH
ret z
call print
inc hl
jr setup1
setup_str: db 13,10,27,"A".8,0FFH ; co 8 pikseli

info: ld hl,info_str : znak firmowy
jr setup1
info_str: db 13,10," "
db "Screen Copy (c)"
db "1990 M. SZOKOL".8,"/O"
db ' & "BAJTEK" '.13,10,0FFH

margines: ld hl,margin : margines 5 spacji
jr setup1
margin: db 13,10," " .0FFH

reset: ld hl,p_reset : reset drukarki
jr setup1
p_reset: db 12,27,"@".0FFH

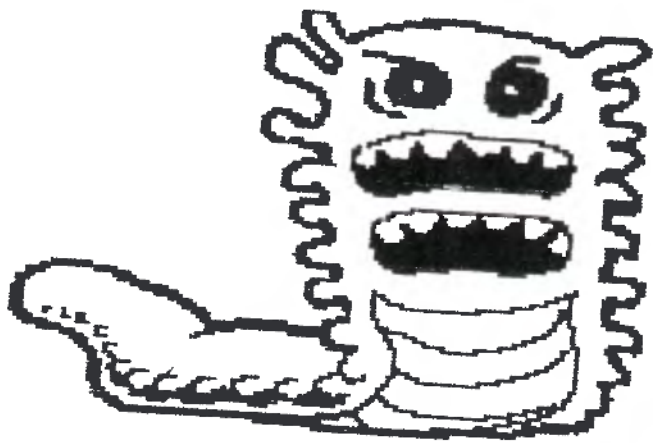
linia: ld hl,set_lin : ustaw parametry paska
jr setup1
set_lin: db 27,"Z" : początkowa rozdzielcz.
dw size*8 : 1600 punktów
db 0FFH

drukuj: call info : przedstaw się
call setup : ustaw wysuw
drukuj1: call margines : petla dla paska
call linia : margines i parametry
ld hl,(ekran) : weź adres na ekranie
ld b,size : długość paska
ld a,(hl) : petla wysłania paska
rept 8
call print : każdy bajt 8 razy
push bc
call sygnal : sygnalizuj pracę
call 0BC29H : nast. bajt - linia w
pop bc : górze!
djnz druk : powtarzaj "size" razy
ld hl,(ekran)
call 0BC20H : nast. pasek - bajt w
ld (ekran),hl : prawo. zapamiętaj
ld a,(licznik)
dec a : zmniejsz licznik
ld (licznik),a
or a : czy zostały jeszcze
jr nz,drukuj1 : paski? TAK=druk nast.
call margines : nowa linia
call reset : reset i nowa strona
ld a,7
call 0BB5AH : dzyń! fajrant!
ld sp,(sav_sp) : odtwórz stos syst.
reti : wróć do systemu

ekran: dw 0 : zmienne
licznik: db 0
sav_sp equ 83FEH : wsk. stosu syst.
int_block: ds 9 : blok przerwania

ph_len equ $-przerwanie : długość segmentu
.dephase : koniec kodu rezyd.
end

```



**Od czasu do czasu świat obiega sensacyjna wiadomość — w komputerze siedzi wirus i zżera dane. Laicy wyobrażają sobie małe zwierzątko — coś pośredniego między myszą a gonokokiem — siedzącego w środku w obudowie i systematycznie, kawałek po kawałku, wygryzające dziury w dyskietkach. Niełaicy na ogół wiedzą, że wirus to taki program, który może skasować coś na dysku albo grać co kwadrans „Panie Janie...” (Frere Jacques). A jak jest naprawdę?**

Przegląd wiedzy o wirusach zaczniemy od samego początku, czyli od pytania:

## CO TO JEST?

Gdyby starać się o ścisłą definicję, wyglądałaby ona mniej więcej tak: wirus komputerowy to taki program, który potrafi się rozmnażać, czyli powielić liczbę „żywych” kopii siebie samego. Żeby program mógł się powielić, musi najpierw zostać uruchomiony, toteż wirusy komputerowe muszą mieć jakiegoś nosiciela, bez którego nie są w stanie nic zrobić. Nosicielem może być albo program, albo dyskietka — nawet taka, na której pozornie nic nie ma. W pierwszym wypadku wirus uaktywnia się w momencie uruchomienia programu, w drugim — po wystartowaniu komputera z zakażonej dyskietki.

Jak to możliwe, że wirusy się powielają? Zwykle zainfekowany już program jest zmodyfikowany w taki sposób, że zanim zacznie się sam wykonywać, wykonywany jest program wirusa. Wirus, korzystając ze standardowo dostępnych w systemie operacyjnym środków (różne przerwania DOS-u), znajduje na dysku inny program, dopisuje do niego swój kod i skok do jego początku, po czym przekazuje sterowanie do nosiciela. Przy wywołaniu zainfekowanego programu operacja powtórzy się — i w ten sposób wirus będzie roznosić się coraz dalej. Jeżeli zainfekowany zbiór sko-

piuje od nas kolega, wirus zacznie się panoszyć u niego i tak dalej — „Przed godziną jedenastą już kichało całe miasto...”

## ILE TEGO JEST?

Dotychczas (według listy opublikowanej 8 sierpnia w USA) wykryto 143 podstawowe typy wirusów, występujące w 213 odmianach. Wiele z nich powstało na podstawie wirusów istniejących już wcześniej — zmieniono tylko pewne niewielkie ich fragmenty, wprowadzając dodatkowe cechy wirusa, nie ruszając części podstawowej, odpowiedzialnej za powielanie. Ponad wszelką wątpliwość oryginalnych wirusów jest jednak więcej niż 143 — są programiści, którzy napisali własne wirusy, lecz nigdy nie wypuścili ich w świat; ponadto amerykańska lista nie zawiera nowych wirusów, powstających ostatnio u naszego wschodniego sąsiada.

## CO POTRAFIĄ WIRUSY?

Prościej byłoby powiedzieć czego nie potrafią — jak na razie nie są w stanie (i można na szczęście liczyć, że ten stan utrzyma się nieco dłużej) zaatakować wyłączonego komputera ani zainfekować dyskietki z załepionym wycięciem. A co potrafią?

Oprócz zdolności do powielania się, wirusy zwykle wyposażone są w ja-

kieś dodatkowe, zwykle destrukcyjne umiejętności — mogą kasować FAT (np. wirus **June 16<sup>th</sup>**, czyli 16 czerwca — robi to właśnie tego dnia), mogą mazać w losowych miejscach dysku (**Datacrime**), zmieniać `command.com` w taki sposób, że przestaje działać instrukcja `copy` (**Vcomm**), grać różne melodie, spowalniać komputer i tak dalej. W porównaniu jednak z tym, co niektóre z nich potrafią robić, by nie dać się złapać, są to czynności prymitywne, jak tłuczenie młotkiem chińskiej porcelany.

Większość nowych umiejętności wirusów ma swoje źródło w programach zabezpieczających przed nimi. W starych dobrych czasach, kiedy komputery były jeszcze mało znane, a wirusów było dwa albo trzy, potrafiły one w najlepszym wypadku przykleić się do zbiorów typu **\*.com**. Bardzo szybko odkryto, że aby uniknąć zarażenia programu, wystarczy mu ustawić atrybut `ReadOnly`. W kilka miesięcy później pojawiły się wirusy, które potrafiły to ominąć, i zarażały programy niezależnie od tego, jak ustawione były atrybuty. Następnym krokiem były więc programy rezydentne, nie pozwalające bez zgody operatora na otwarcie (za pośrednictwem systemu operacyjnego) zbiorów typu **\*.com** i **\*.exe**. W kilka miesięcy później pojawiły się wirusy, które potrafiły i te zabezpieczenia obejść.

Od tamtego czasu upłynęło parę lat i kilka osób zrobiło duże pieniądze na programach zabezpieczających przed własnymi wirusami do komputera — zdobywały nowe sprawności. Nauczyły się nawet wykorzystywać niektóre ze stosowanych w programach antywirusowych chwytów do własnych zbójczych celów. Koronnym przykładem jest jeden z najpaskudniejszych wirusów — **Dark Avenger** (który mimo angielskiej nazwy powstał w Bułgarii). Jego podstawowy sposób działania nie jest niczym niezwykłym — po uruchomieniu zainfekowanego programu wirus instaluje się rezydentnie w systemie i czeka (robi to wiele wirusów). Perfidia jego działania pole-

ga na tym, że zarażony zostaje każdy zbiór typu **\*.com** lub **\*.exe** używany przez system — czy to w celu uruchomienia, czy skopiowania — toteż jeżeli **Dark Avenger** tkwi w pamięci, a ktoś używa programu antywirusowego, sprawdzającego wszystkie programy na dysku, wszystkie zostaną zarażone niejako na własną prośbę!

**Dark Avenger** jeszcze pod jednym względem zalicza się do „najzdolniejszych” wirusów — przy wywoływaniu BIOS-u w celu wykonania potrzebnych mu operacji na dysku zamiast korzystać z przerwań skacze bezpośrednio pod potrzebny adres. Dzięki temu może on ominąć wszelkie programy podobne do opisanego dziś w klanie SETDP 1/2.

Wirusy na ogół rozpoznawane są przez programy antywirusowe na podstawie fragmentów ich kodu (w programie wykrywającym i leczącym z **Yankee Doodle** taki fragment jest zapamiętany w tablicy **ident**. Pierwszym sposobem na uniknięcie identyfikacji było kodowanie siebie — potrafiły to wirusy **1704/1701**. Nie jest to zbyt skuteczna metoda, bo sama procedura dekodująca była zawsze taka sama i można ją było wykryć. Ostatnio pojawiły się wirusy (**COM\_AIDS**, **ZIP** i **Virus-101**), które za każdym razem przy powielaniu siebie tworzą nową, praktycznie niepowtarzalną procedurę dekodującą — ich rozpoznanie stało się przez to niezwykle trudne (na szczęście nie niemożliwe).

Co więc może umieć wirus, który próbuje wskoczyć na nasz dysk? Może ominąć wszystkie programy blokujące mu dostęp, zaszyfrować się tak, że nie będziemy go w stanie zidentyfikować, zmodyfikować system w taki sposób, że nie będziemy w stanie wykryć zmian długości zarażonych programów (np. **4096** i **Joshi**), zainstalować się rezydentnie w systemie, atakować każdy uruchamiany zbiór i każdą wkładaną do stacji dyskietkę, a na dodatek — jakby tego wszystkiego było mało — może formatowanie zainfekowanego dysku przecześć w pamięci operacyjnej i wskoczyć nań z powrotem (**Stoned**), kiedy zadowolony z siebie przystąpimy do odtwarzania zawartości dysku. Może skasować FAT, tablicę partycji dysku, może zmieniać losowo wybrane bajty zapamiętane na dysku, grzebać w zbiorach **\*.dbf** tworzonych przez `dBase`, zamieniać miejscami cyfry w czterocyfrowych liczbach wyświetlanych na ekranie, grać niemieckie melodie ludowe, wypisywać bzdurne komunikaty, spowalniać pracę komputera i tak dalej, tak dalej, tak dalej...

## JAK SIĘ PILNOWAĆ?

Najpewniejszym sposobem na uniknięcie infekcji jest wyłączenie komputera i zapakowanie go do mocnego kartonowego pudła. Skoro jednak już decydujemy się na korzystanie z komputera, trzeba przyjąć kilka reguł postępowania. Całkowicie wyeliminować ryzyka się nie da, ale tkając odpowiednio gęstą sieć, poważnie zmniejszymy szanse wirusa na zainfekowanie dysku w sposób niezauważalny dla nas.

Najważniejsze przykazanie brzmi tak: nie wsadź do stacji każdej dyskietki, jaką uda ci się zdobyć i nie uruchamiaj na swoim komputerze każdego programu, który znajdziesz. Jeżeli już chcesz sprawdzić coś nowego pamiętaj że:

1. Wirus będzie próbować zarazić twardego dysk lub znajdujące się na nim programy i nie ma żadnego pewnego sposobu na uniemożliwienie mu tego na drodze programowej. Dlatego też, o ile możesz to zrobić, odłącz fizycznie twardego dysk od reszty komputera (wystarczy odłączyć kabel od sterownika).
2. Wirus może być rezydentny, toteż po zakończeniu testowania programu należy komputer wyłączyć i wyłączyć z sieci, aby mieć absolutną pewność, że nic nie zostało w pamięci. Można go również zresetować, ale guzikiem z napisem Reset, a nie kombinacją ctrl-alt-del, na którą tylko czeka **Alameda!**
3. Przed uruchomieniem nowego programu warto wgrać jakiś program zabezpieczający przed replikacją wirusa — może to być **flus-hot** albo **virblock**, albo jakiś ich kuzyn. Istnieje wprawdzie sporo wirusów, które potrafi ominąć te programy, ale wiele tego nie potrafi i zostaną dzięki temu zdemaskowane.
4. Przed skopiowaniem przetestowanego programu na twardego dysk dobrze jest poczekać kilka dni lub tygodni — może wirus ujawni się w tym czasie u kogoś innego?
5. Na wszelki wypadek dobrze mieć w systemie kilka programów sprawdzających co jakiś czas stan „sanitarno-epidemiologiczny” dysku — mogą to mogą być nawet programy najprostsze, takie jak prezentowany w klanie **comcheck** — mimo iż każdy z nich może zostać oszukany, zmniejszają one pole manewru wirusa.

Oczywiście staranność w przestrzeganiu powyższych reguł zależy od użytkownika — jeżeli nie chce, nie musi ich stosować. Wiąże się to z ryzykiem utraty wszystkich danych, ale że koszt zabezpieczeń nie powinien przekraczać wartości zabezpieczanych danych, dysku z samymi gramami nie warto pilnować zbyt pedantycznie. Trzeba jednak pamiętać o tym, że każdy wirus, któremu zostawi się swobodę, prędzej czy później będzie próbował zaatakować następny dysk u kogoś innego.

## JAK SIĘ LECZYĆ?

Jeżeli już zdarzy ci się znaleźć na dyskietce wirusa, przede wszystkim

nie wpadaj w panikę. Najczęstszym błędem popełnianym w takiej sytuacji jest natychmiastowe formatowanie zainfekowanej dyskietki (lub, co jeszcze gorsze, twardego dysku). Jest to błąd z trzech powodów. Po pierwsze, nie daje to żadnej gwarancji pozbycia się wirusa na stałe (**Stoned**). Po drugie, oznacza to nieodwracalną utratę wszystkich danych — ponieważ nie grozi nam już nic gorszego, można spróbować je uratować, ryzykując utratę tylko ich części. Po trzecie wreszcie, każdego wirusa należy złapać, zidentyfikować i zbadać, aby móc się przed nim bronić w przyszłości. Jeśli sam nie dysponujesz takimi umiejętnościami, poproś o pomoc kogoś, kto się na tym zna (**jeżeli nie znasz nikogo takiego, spróbuj skontaktować się z nami**).

Jakie są najprostsze działania, które można podjąć samemu? Przede wszystkim należy wystartować system z absolutnie pewnej zaklejonej dyskietki — trzeba mieć przygotowaną taką na wszelki wypadek, zawierającą oprócz podstawowych zbiorów systemowych programy antywirusowe i program typu **Norton Disk Doctor**, **Norton Utilities** albo **PCTools**. Następnie trzeba spróbować wykryć sprawcę kłopotów za pomocą jakiegokolwiek programu szperającego po dysku — musi to być jednak dobry program, im nowszy, tym lepszy (na przykład — według stanu z połowy września 1990 — **scanv66b**, albo **MkS\_vir** wersja 2.94, opisany obok). Być może szperacz będzie w stanie wyleczyć zainfekowane programy — można mu na to pozwolić, ale niezależnie od ewentualnego komunikatu, że dysk jest już zdrowy jak ryba, radzę jeszcze raz powtórzyć całą operację, od startowania z pewnej dyskietki po komunikat o pełnym zdrowiu — może się zdarzyć, że programy były zainfekowane wielokrotnie, a nie każdy program antywirusowy potrafi w jednym przebiegu wyleczyć ten sam zbiór kilka razy. Jeżeli po tych działaniach nadal dzieją się dziwne rzeczy i podejrzewasz (lub masz pewność), że wirus dalej grasuje po dysku — zawołaj kogoś, kto się na tym zna.

Jeżeli zdecydujesz się na formatowanie, radzę zrobić to w nieco nietypowy sposób (korzystając wyłącznie z pewnej zalepionej dyskietki systemowej, nigdy z twardego dysku, na którym siedzą wirusy!). Po wystartowaniu systemu i uruchomieniu programu formatującego, zanim skończy on swoją pracę, wyłącz na sekundę komputer z sieci, po czym sformatuj dysk od początku. W ten sposób każdy wirus, który próbuje przeczekać formatowanie w pamięci operacyjnej, dostanie w łeb. Sekundowa przerwa w dopływie prądu jest zbyt krótka, by dysk mógł ulec uszkodzeniu, mimo że nie został zaparkowany. (Ktoś może mi zarzucić, że po wystartowaniu systemu z pewnej dyskietki resetowanie systemu nie jest potrzebne — wiem o tym, ale ostrożności nigdy za wiele).

Powyższych rad, zarówno tych dotyczących ochrony przed wirusami, jak i leczenia zainfekowanych programów i dysków, nie należy traktować w sposób ostateczny. Doświadczenie uczy bowiem, że w tej dziedzinie każda informacja wychodząca na światło dzienne jest już przynajmniej częściowo nieaktualna. Dlatego też od czasu do czasu będziemy wracać do tego tematu.

*Marcin Borkowski*

# MkS\_vir

Z warszawskiej firmy **UNIKS** dostaliśmy do testowania program antywirusowy **MkS\_vir**. Program był nam już znany dzięki wersji demonstracyjnej, krążącej po Polsce. Mają one jednak ograniczone możliwości — potrafią tylko wykryć różne wirusy, nie lecząc zarażonych nimi zbiorów. Wersja licencjonowana w większości wypadków oprócz diagnostyki potrafi również przeprowadzić skuteczną kurację.

### PIERWSZY RZUT OKA

Program robi dobre wrażenie — stara się być przyjacielski i podpowiada co robić, żeby wywołać określone akcje. Dołączona broszurka (znajdująca się również w postaci pliku z Chi-Writer-a na dyskietce dystrybucyjnej) pozwala jeszcze przed uruchomieniem programu zorientować się w zasadach jego działania, i — co ważne — podaje podstawowe wiadomości na temat wirusów i zasad „higieny”, których przestrzeganie zmniejsza ryzyko. Broszurka napisana jest dość uczciwie — nie stara się wmówić nikomu, że **MkS\_vir** potrafi więcej niż potrafi w rzeczywistości, ale też brak w niej jasnego postawienia sprawy — wobec nowych wirusów program jest bezradny.

Wypróbowałem program na kilku wirusach, które znajdują się w redakcyjnych zbiorach — z pozytywnym skutkiem. Wszystkie zakażone zbiory były identyfikowane i leczone, chociaż nie zawsze w sposób dający się zaakceptować. Po uruchomieniu programu w systemie zainfekowanym przez **Yankee Doodle**, **MkS\_vir** najpierw obwieścił, że znalazł YD w pamięci i go unieszkodliwił; potem, że YD jest w pamięci i nie potrafi sobie z nim poradzić. Prawdziwe okazało się być pierwsze stwierdzenie — YD był unieszkodliwiony. Drugi raz udało się **MkS\_vir**-a wywieść w pole za pomocą programu zainfekowanego dwukrotnie — przez **648** i YD. Po eleganckim wykryciu i usunięciu YD na ekranie poja-

wiły się komunikaty o dziwnych wirusach, którymi jeszcze jest zarażony program — miały to być **Vienna A** i **Lisbon**. Niczym nie zrażony, kazałem sprawdzić badane zbiory jeszcze raz i tym razem **648** został usunięty już bez trudu.

### DRUGI RZUT OKA

Wrażenie jest już trochę gorsze, z powodów opisanych powyżej — pojawiające się kolejno po sobie sprzeczne komunikaty i wykrywanie nie istniejących wirusów mogą kogoś mało wprawnego doprowadzić do rozpaczy. Niemniej jednak **program wykonuje swoje zadanie skutecznie i szybko, co czyni z niego wygodne narzędzie walki z wirusami**. Jeżeli w trakcie badania dysku zostały wykryte jakieś wirusy, warto po ich wyleczeniu na wszelki wypadek przejrzeć dysk jeszcze raz.

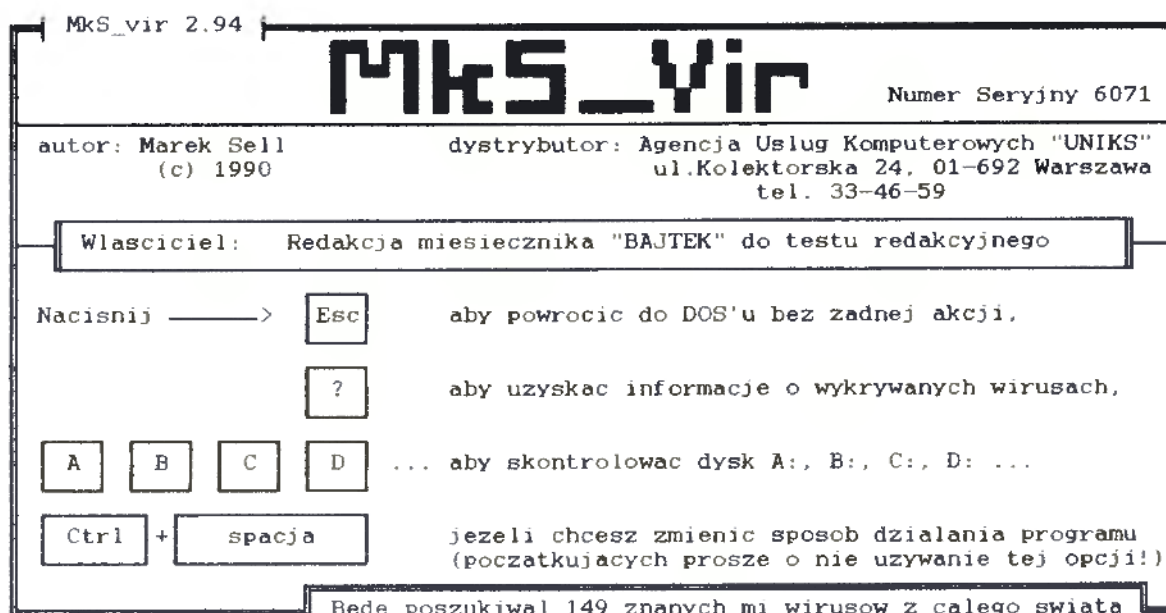
### JESZCZE KILKA SZCZEGÓŁÓW

Oprócz szukania i leczenia zainfekowanych programów, **MkS\_vir** zawiera opisy piętnastu wirusów, razem z demonstracjami efektów przez nie wywoływanych. Program można dopasować do kilku różnych standardów polskich liter — min. DHN, CSK, Mazovia, Latin-2 — albo zrezygnować z nich zupełnie. Zasadniczo **MkS\_vir** przeznaczony jest do pracy interaktywnej, ale można go również umieścić w pliku wsadowym, podając jako parametry dyski, które mają być sprawdzone.

Testowana przez nas we wrześniu wersja ma numer 2.94 — zanim „Bajtek” 11-12/90 pojawi się w kioskach, będą już nowe wersje programu. Średnio co półtora do dwóch miesięcy pojawia się nowa wersja, zwalczająca oprócz starych dwa nowe wirusy. Posiadacze starszych wersji mogą zostać właścicielami nowych po obniżonej cenie. Obecnie program rozpoznaje 149 wirusów, zwalcza 42 plus ich różne mutacje.

*Marcin Borkowski*

Producent programu:  
Agencja Usług Komputerowych **UNIKS**  
telefon 33-46-59  
ul. Kolektorska 24  
01-692 Warszawa  
Autorem programu jest pan Marek Sell

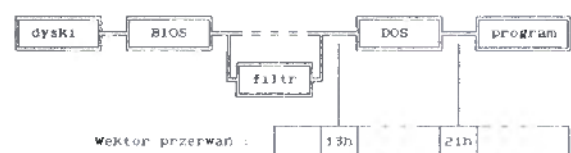


# OCHRONA DYSKÓW TWARDYCH PRZED WIRUSAMI

**W artykule przedstawiono program rezydentny blokujący możliwość zapisu na dysku sztywnym. W zasadzie jest to para programów przygotowana z myślą o komputerach IBM-PC wyposażonych w dwa dyski twarde, ale można je wykorzystać również w konfiguracji z jednym dyskiem.**

Jeżeli posiadamy dwa dyski, to wygodne jest przyjęcie reguły, że na pierwszym dysku przechowywane są: system operacyjny i całe posiadane oprogramowanie użytkowe, drugi zaś służy do pamiętania plików użytkownika, zawierających dane, teksty, wersje źródłowe opracowywanych samemu programów itp. Celowe jest wtedy zablokowanie możliwości zmian na pierwszym dysku w taki sposób, by maksymalnie utrudnić atak wirusa. Pliki z drugiego dysku nie są tak bardzo zagrożone jego atakiem, gdyż jeśli nawet przyklei się do wersji binarnych pisanych przez nas programów, to dzięki ich częstej kompilacji zostanie szybko zniszczony.

Poniższy rysunek przedstawia sposób realizacji blokady. Sprowadza się ona do zainstalowania pomiędzy warstwy BIOS-u i DOS-u filtru, który nie przepuszcza do BIOS-u żądań zapisu na dysk twarde. Po zainstalowaniu filtru można zmieniać jego działanie, co pozwala na włączanie i wyłączanie blokady dla jednego lub obu dysków bez potrzeby restartowania systemu.



Wszystkie operacje dyskowe — niezależnie, czy inicjowane z poziomu komend DOS-u, programów użytkowych czy też wirusów — sprowadzane są przez system operacyjny do ciągu elementarnych operacji dyskowych, polegających na odczycie i za-

pisie wskazanych sektorów dyskowych. Operacja ta jest zlecona BIOS-owi i po drodze natrafia na zainstalowany filtr, który — jeśli uzna żądanie za legalne — przekazuje je do adresata, jeśli nie — symuluje powrót z BIOS-u z informacją o błędzie zapisu. W efekcie dysk twarde zachowuje się jak dyskietka z zaklejonym wycięciem — wszystkie próby zmiany zawartości kończą się komunikatem:

„Write protect error writing drive ...”

Wirusy rozmnażają się na dyskach z reguły za pośrednictwem funkcji DOS-u, swoje zaś niszczyielskie cele mogą realizować również na poziomie BIOS-u. W obu przypadkach ich działania zostaną zablokowane przez filtr, chyba że będziemy mieć do czynienia z wirusem „znającym” adres potrzebnej mu procedury w BIOS-ie.

Do realizacji blokady służą programy SETDP1 i SETDP2 (Set Disk Protection). Przy pierwszym wywołaniu program SETDP1 instaluje filtr i pozostawia go w stanie blokującym dostęp czynny do pierwszego dysku twardego i umożliwiającym zmiany na dysku drugim. Kolejne wywołania tego programu powodują na przemian zdejmowanie blokady z obu dysków i zakładanie blokady na dysk 1, co zostaje potwierdzone odpowiednim komunikatem. Program SETDP2 również potrafi zainstalować filtr, jednak jego głównym celem jest naprzemienne nakładanie blokady na oba dyski i zdejmowanie blokady z dysku 2.

Program SETDP1 przeznaczony jest do wpisania do pliku AUTOEXEC. Po załadowaniu systemu mamy wtedy zablokowany dysk 1 (z oprogramowaniem) i dostępny dysk 2 (użytkowy). Przy typowej pracy nie ma potrzeby zmieniania stanu blokady. Jeśli jednak chcemy zapisać coś na dysku 1, to odblokowujemy go komendą SETDP1. Jej ponowne użycie przywraca blokadę dysku 1. Ponadto przed wczytaniem z dyskietki programu z niewiadomego źródła możemy na wszelki wypadek zablokować komendą SETDP2 również drugi dysk. Jej ponowne użycie odblokuje dysk 2, pozostawiając zablokowany dysk 1. Jeżeli posiadamy tylko jeden dysk lub nie zamierzamy blokować

```

title SETDP1
page 66,132
segment
assume cs:cseg
org 100h
start: jmp init
oldint13 dd ?
perm1 db 0fh ; zgoda na zapis na dysk 1
perm2 db 0fh ; zgoda na zapis na dysk 2
newint13 proc far ; procedura filtrująca
cmp ah,03h ; zadanie zapisu sektora?
je check
cmp ah,05h ; zadanie formatowania ścieżki?
jb continue
cmp ah,07h
jbe check
cmp ah,0bh ; zadanie 'długiego' zapisu?
je check

```

```

continue: jmp cs:[oldint13] ; przekaz sterowanie do BIOS-u
check : cmp di,80h ; zadanie zapisu na dysk 1?
je hd1
cmp di,81h ; zadanie zapisu na dysk 2?
je hd2
jmp continue
hd1 : cmp perm1,0fh ; czy jest zgoda na zapis
je continue ; na dysk 1?
jmp abort
hd2 : cmp perm2,0fh ; czy jest zgoda na zapis
je continue ; na dysk 2?
abort: mov ah,03h ; 3 oznacza nieudana próbę
; zapisu na zabezpieczony dysk
; CARRY flag sygnalizuje błąd
; podczas operacji
stc
ret 2
newint13 endp ; koniec procedury filtrującej
init : ; przeszukiwanie pamięci
mov dx,offset newint13
mov ax,cs
mov es,ax
nextpar: dec ax
mov ds,ax ; DS:SI - badany paragraf
mov si,dx ; ES:DI - wzorzec
mov cx,10
cld
repe cmpsw
jnz noteq
cmp ds:perm1,0fh
jnz switch
noteq : cmp ax,0001h
jnz nextpar
notfound : ; instalacja filtru
mov ax,3513h ; pobierz adres oryginalnej
int 21h ; funkcji BIOS-u
word ptr cs:[oldint13],bx
word ptr cs:[oldint13 + 2],es
push cs
pop ds
mov dx,offset newint13
mov ax,2513h ; wpisz adres newint13 do
int 21h ; wektora przerwań
mov perm1,0 ; ustaw przełączniki
mov perm2,0fh
mov dx,offset protect_on
mov ah,9
int 21h ; wyświetl komunikat
mov dx,offset init
int 27h ; zostaw newint13 w pamięci
switch: not ds:perm1 ; zmień stan przełączników
cmp ds:perm1,0
jz on
mov ds:perm2,0fh
mov dx,offset protect_off
jmp exit
on: mov dx,offset protect_on
exit: mov ah,09h
push cs
pop ds
int 21h ; wyświetl komunikat
int 20h
protect_on db "Hard Disk 1 Protection ON",13,10,"$"
protect_off db "Hard Disk 1&2 Protection OFF",13,10,"$"
cseg ends
end start

```

```

title SETDP2
..... ( jak wyzej )
mov perm1,0 ; ustaw przełączniki
mov perm2,0
mov dx,offset protect_on
mov ah,9
int 21h ; wyświetl komunikat
mov dx,offset init
int 27h ; zostaw newint13 w pamięci
switch: mov ds:perm1,0
not ds:perm2 ; zmień stan przełączników
cmp ds:perm2,0
jz on
mov ds:perm2,0fh
mov dx,offset protect_off
jmp exit
on: mov dx,offset protect_on
exit: mov ah,09h
push cs
pop ds
int 21h ; wyświetl komunikat
int 20h
protect_on db "Hard Disk 1&2 Protection ON",13,10,"$"
protect_off db "Hard Disk 2 Protection OFF",13,10,"$"
cseg ends
end start

```

# oswajanie PC-eta

dostępu do dysku 2, to program SETDP2 nie jest potrzebny. Program SETDP2 został przedstawiony w skróconej postaci i omówiony wspólnie z SETDP1, gdyż oba programy różnią się jedynie końcową sekwencją instrukcji.

Programy należy doprowadzić do postaci COM w następujący sposób:

- wpisać treść programu SETDP1 do pliku o nazwie setdp1.asm
- skompilować zleceniem : MASM setdp1 (lub TASM);
- zlinkować zleceniem : LINK setdp1 (lub TLINK);  
(zignorować ostrzeżenie „no stack segment”)
- przeformatować zleceniem : EXE2BIN setdp1.exe setdp1.com
- usunąć niepotrzebny już plik setdp1.exe (analogicznie należy postąpić z SETDP2).

Działanie programów rozpoczyna się od przeszukania pamięci (etykieta **init**) w celu sprawdzenia, czy filtr został już wcześniej zainstalowany. Para rejestrów ES:DI wskazuje wzorcowe 10 słów zawierających początek kodu filtru, a para DS:SI jest cyklicznie przestawiana na kolejne paragrafy pamięci. Przeszukiwanie odbywa się z krokiem 16 bajtów, gdyż DOS ładuje programy zaczynając od pełnego paragrafu. Jeżeli filtr nie został znaleziony, to następuje jego instalacja (etykieta **notfound**). Polega ona na zapamiętaniu adresu oryginalnej funkcji BIOS-u, wpisaniu do wektora przerwań adresu funkcji filtrującej **newint13**, ustawieniu przełączników **perm1** i **perm2** decydujących o stanie blokad dla dysków 1 i 2 i pozostawieniu filtru w pamięci na stałe. Jeżeli filtr był zainstalowany wcześniej, to następuje przejście do etykiety **switch**, zmiana stanu przełączników **perm1** i **perm2**, wyświetlenie komunikatu o ich stanie po zmianie i zakończenie programu.

Po zainstalowaniu filtr przechwytywa wszystkie żądania dotyczące operacji na napędach dyskowych. Zlecenia dotyczące dyskietek oraz różne od żądania zapisu są przepuszczane do BIOS-u (skok do adresu zapamiętanego w zmiennej **oldint13**). W przypadku przechwycenia zlecenia numer 3 i Bh (zapisz sektor) lub 5,6,7 (formatuj ścieżkę) sprawdzany jest stan przełącznika odpowiadającego numerowi dysku twardego, którego dotyczy żądanie. Jeżeli przełącznik ma wartość FFh, to zlecenie jest przekazywane do BIOS-u (etykieta **continue**). Jeśli nie, to ustawiana jest flaga CARRY, do rejestru AH wpisywany kod 3 i sterowanie wraca do DOS-u. Z punktu widzenia zleceniodawcy wygląda to tak, jak gdyby operacja nie udała się z powodu sprzętowej blokady zapisu.

Należy zaznaczyć, że przez dysk 1 i 2 BIOS rozumie numer dysku fizycznego. Jeśli dysk jest podzielony na partycje widziane przez DOS jako różne dyski logiczne, to na poziomie BIOS-u są one tym samym dyskiem fizycznym. Przełączniki **perm1** i **perm2** sterują więc dostępem do wszystkich partycji danego dysku.

Stosowanie blokady bardzo nieznacznie uszczupla pamięć operacyjną komputera, gdyż instalowana na stałe jest jedynie funkcja **newint13**. Po zainstalowaniu zajmuje ona 336 bajtów (wraz z PSP). Do tego dochodzi jeszcze dziedzićzona przez program kopia opisu środowiska (wielkość rzędu 60-200 bajtów). Blokada praktycznie nie spowalnia legalnych operacji na dyskach.

Marek T. Frydryszak

... nie jest wcale trudne. Najpierw trzeba go włączyć. Aby móc to zrobić, musisz mieć pewność, że:

1. Komputer jest włączony do sieci.
2. Klawiatura i monitor są podłączone przewodami do jednostki centralnej.
3. Komputer jest wyposażony w twardego dysku lub w stacji A: (górnej lub jedynej) znajduje się dyskietka systemowa.

Jeśli wszystko jest gotowe — zaczynamy. Wyłącznik znajduje się na ogół z boku lub z tyłu. Pstryk. Jeżeli w ciągu kilkudziesięciu sekund na ekranie (czy monitor jest włączony?) nie pojawi się znak zachęty (zakończony zwykle znakiem >), musisz szukać dyskietki systemowej.

## UWAGI OGÓLNE

które przeczytać powinni wszyscy. Na dysku lub dyskietce znajdują się zwykle różne zbiory i katalogi. Katalog to taka szuflada, w której są zgromadzone różne zbiory, według czyjegoś widzimisię; może w niej być również podkatalog, czyli następna szuflada, i tak dalej. Każdy katalog i każdy zbiór mają swoją nazwę. Nazwy zbiorów zwykle są dwuczęściowe — składają się z samej nazwy i z rozszerzenia,

oddzielonych kropką. Nazwa to nazwa, a rozszerzenie mówi o tym, co w danym zbiorze się znajduje. Najważniejsze dla nas rozszerzenia to **.com**, **.bat** i **.exe** — zbiory o tych rozszerzeniach to programy, które można uruchomić. Nazwy katalogów na ogół nie mają rozszerzenia. Polecenia wydaje się pisząc je i naciskając klawisz Enter. W razie jakichś trudności z odzyskaniem kontroli nad komputerem można nacisnąć trzy klawisze naraz — **Ctrl-Alt-Del**. Powinno to spowodować ponowne uruchomienie systemu.

## TWARDY DYSK

różni się od dyskietki tylko tym, że jest szybszy, mieści się na nim więcej zbiorów i jest przykręcony na stałe gdzieś w środku. Na ogół po uruchomieniu komputera z twardego dyskiem na ekranie pojawiają się dwie duże prostokątne ramki, zawierające listy zbiorów i katalogów. Znaczący to, że działa specjalny program ułatwiający pracę (Norton Commander). Jeśli go nie ma — przeczytaj to, co jest napisane o systemie bez twardego dysku.

W okienku Nortona widać nazwy wszystkich zbiorów i katalogów. Jedną z nich jest podświetlona. Jeśli jest to nazwa programu (rozszerzenie!), naciśnięcie klawisza Enter spowoduje

wykonanie tego programu, jeśli jest to podkatalog (**SUB-DIR**) — wejście do niego, jeśli katalog macierzysty (**UP-DIR**) — powrót. Można również wydawać za pomocą klawiatury wszystkie polecenia opisane poniżej (Tylko dyskietki). Litery mogą być duże lub małe, nie ma to znaczenia.

**Uwaga!** komputer z twardego dyskiem trzeba „zaparkować” poleceniem **park** lub **bye** przed wyłączeniem!

## TYLKO DYSKIETKI

spotyka się coraz rzadziej, ale warto wiedzieć, jak sobie dać radę. Żeby sprawdzić, co jest na dyskietce, trzeba wydać komendę **dir**. Na ekranie pojawi się lista wszystkich zbiorów i podkatalogów. Jeżeli lista jest za długa i ucieka zanim zdążysz ją przeczytać, napisz **dir /p**. Podkatalogi i katalogi macierzyste zaznaczone są napisem <DIR>. Do podkatalogu przechodzi się za pomocą instrukcji **cd nazwa**, do katalogu macierzystego **cd ..** (przerwa przed kropkami mile widziana, choć nie zawsze konieczna). Programy uruchamia się tak samo jak wydaje polecenia — pisząc nazwę i naciskając Enter

Marcin Borkowski

STRZEŻ SIĘ WIRUSÓW...

**Łatwo powiedzieć, trudniej zrealizować. Wirus to taki program, który udaje, że go wcale nie ma, „podpełźnie cichaczem, niby nie wiedzieć za czym, tylnie z czeluści, korzonek zapuści i jak potem da łupnia...” Trzeba szukać sposobów na to, by uniknąć kłopotów. Można w tym celu stosować różne P(r)AW-ie skuteczne programy sprawdzające, leczące, blokujące i utrudniające. Można (a nawet trzeba) bardzo starannie unikać przygodnych dyskietek, a przynajmniej zabezpieczać się przed infekcją stosując odpowiednie środki zaradcze.**

Wszystko to może zmniejszyć ryzyko złapania czegoś przez nasz dysk, zwykle za cenę spowolnienia szybkości komputera lub poważnego ograniczenia wydajności. Niestety, nie ma sposobów jedynek i zawsze skutecznych — prędzej czy później trafi się wirus, który bez problemu ominie zabezpieczenia. Szczęśliwie dla nas większość wirusów (ale nie wszystkie) infekując twardego dysku atakuje w pierwszym rzędzie **command.com**, zmieniając przy tym jego długość (to też nie jest do końca prawdą). Dzięki temu bardzo prostym, a przy okazji dość skutecznym, testem stanu dysku jest częste sprawdzanie długości **command.com**-a. Żeby nie musieć o tym pamiętać, warto napisać krótki program i umieścić go w **autoexec**-u. Nie zabezpiecza to bynajmniej przed wtargnięciem wirusa do systemu, ale zwiększa szanse na jego szybkie wykrycie, i wczesne podjęcie środków zaradczych.

Program taki może wyglądać tak jak prezentowany **com-check.pas**. Przedstawiona wersja jest przeznaczona dla systemu w wersji 3.30, ale wystarczy odpowiednio zamienić liczbę 25307 (długość **command.com**-a DOS-u 3.30), by program mógł pracować z dowolną wersją DOS-u.

Oczywiście istnieje sporo wirusów, które postępują w sposób na tyle inteligentny, że ominą próbę ich wykrycia — niektóre z nich w ogóle nie ruszają programów, a mnożą się na samych dyskach, inne potrafią zainfekować **command.com** nie zmieniając jego długości (korzystając ze

zdefiniowanego wewnątrz programu obszaru przeznaczonego na stos), jeszcze inne nie infekują właśnie **command.com**-a, lecz inne programy. Można więc prezentowany program nieco rozbudować, dodając na przykład konieczność sprawdzenia własnego rozmiaru lub kilku dowolnych innych programów. Pozostawiam to czytelnikom, proponując im na razie najprostszą wersję, która już dwa razy okazała się zbawienna — wykryła infekcję zanim wirusy zdążyły się na dobre zdomowić na dysku.

Marek Ciężarek

```

uses crt,dos;

var a : file of byte;

procedure beep(i : integer);
begin
  sound(i); delay(1)
end;

procedure syrena;
var i : integer;
begin
  repeat
    for i:=300 to 800 do beep(i);
    for i:=800 downto 300 do beep(i);
  until keypressed
end;

begin
  assign(a,getenv('comspec'));
  filemode:=0; reset(a);
  if filesize(a)>>25307 then { 3.30 }
  begin
    highvideo;
    writeln(#13#13#10'Uwaga! Wirus!');
    lowvideo;
    syrena
  end;
  nosound; close(a)
end.

```

# Yankee Doodle

Kilka tygodni temu na jednej z moich dyskietek zagnieździł się wirus. Jak się na niej znalazł, to osobna historia, której nie udało się do końca wyjaśnić i — według wszelkiego prawdopodobieństwa — nie uda się już nigdy. Wirus na dyskietce to żadna przyjemność, ale kilka lat pracy z PC-tem nauczyło mnie spokoju i oparowania w takich sytuacjach, więc i tym razem bez wpadania w panikę postanowiłem spróbować oparować sytuację.

Pierwszy etap to próba zidentyfikowania wirusa — która się powiodła. Udało mi się z pomocą programu **scan** ustalić nazwę wirusa — Yankee Doodle, ale okazało się, że nie mam pod ręką szczepionki, która potrafiłaby sobie z nim poradzić. Dowiedziałem się jeszcze (po przekopaniu kilkunastu artykułów), że jest to wirus rezydentny, że przykleja się do zbiorów **\*.exe** i **\*.com** (co już i tak wiedziałem), że bardzo nie lubi być podglądany i że od czasu do czasu gra melodię Yankee Doodle. Kilkogodzinna obserwacja wirusa (na XT, AT i 386 — na tym ostatnim komputerze próba infekcji nieodmiennie kończyła się zawieszeniem systemu) nie wykazała żadnych jego groźnych cech, usunięcie zakażonych plików (przez zwykłe del) rozwiązywało więc sprawę definitywnie, gdyby nie jeden drobny fakt. Zaa-takowany został program, którego nie chciałem się pozbywać. Zmusiło mnie to do obejrzenia wirusa w akcji, pod kontrolą odpluskiwacza, i do starannego przeanalizowania zawartości zainfekowanych plików. W trakcie tych prób okazało się, że wirus nie

lubi się z programem **dosedit**, a konkretnie z oferowanymi przez ten program „zastępstwami” (alias), które giną podczas infekowania systemu. Nie było łatwo, ale koniec końców udało mi się zdobyć wystarczającą ilość informacji, by móc napisać program leczący.

Program ten szuka ciągu ośmiu charakterystycznych bajtów, które w przypadku **exe** leżą 880 bajtów przed końcem programu, w przypadku **com** cztery bajty wcześniej. Jeżeli zostaną one znalezione, program jest zainfekowany. Leczenie polega na przeniesieniu 32 bajtów zapamiętanych na początku wirusa (koniec pliku minus 2871) na początek pliku, skąd pochodzą, i amputacji samego wirusa, czyli tego, co zostało do programu dopisane. Długość programu po infekcji (zapisana w systemie szesnastkowym) kończy się zawsze cyfrą 1 (**exe**) lub 5 (**com**) — długość zainfekowanego programu wzrasta o 2881 do 2895 bajtów. Nie pozwała to na odtworzenie pierwotnej długości zbioru, dlatego też program po wyleczeniu może mieć długość nieco inną niż początkowa — zwykle większą o kilka bajtów. (Być może jest na to jakiś sposób, ja go nie znalazłem). Oparłem się na założeniu, że można obciąć najwyżej ostatnie 2881 bajtów zbioru — jest to minimalny wzrost długości zbioru, jaki widziałem.

W przypadku niektórych wirusów — na przykład **648** — program może służyć jako pomoc przy przygotowywaniu kuracji. Zawiera już wszystkie elementy potrzebne przy leczeniu — identyfikację, odtworzenie oryginalnego programu i amputację wirusa. Oczywiście dokonanie przeróbki wymaga starannego zbadania wirusa i pozmienniania występujących w programie stałych, ale samą konstrukcję — już sprawdzoną — można zostawić.

**Uwaga:** Ponieważ wirus jest rezydentny, leczenie zbiorów ma sens tylko po wystartowaniu systemu z absolutnie pewnej zalepionej dyskietki!

Marcin Borkowski

# ILE DANYCH POTRZEBA NA JEDNĄ BAZĘ?

**Baza danych — to jedno z kilkunastu haseł kojarzonych najczęściej z użytkowaniem komputerów. Spróbujmy wniknąć trochę głębiej w rzeczywistą treść kryjącą się za tym hasłem.**

Moment jest korzystny, bo w kilku ostatnich wydaniach „Bajtki” uczeni ludzie objaśniali techniczną stronę przechowywania danych, bazując na urządzeniach pamięci dyskowych. Możemy więc śmiało wniknąć nie troszcząc się o szczegóły techniczne, poświęcając całą uwagę problemom logicznym.

Aby zachować porządek, zaczniemy od precyzyjnego określenia, jaka informatyczna treść kryje się za obiegowym pojęciem „baza danych”? Tu pozwolę sobie podkraść przykład z artykułu o wykorzystaniu dysków, zamieszczonego niedawno w „Bajtku”.

Stworzono tam, zapisaną w pamięci dyskowej komputera, książkę telefoniczną. (Żeby nie było wątpliwości, ja przez „książkę telefoniczną” rozumiem alfabetyczną listę nazwisk wszystkich abonentów, zawierającą numer telefonu i adres każdego z nich). Czy możemy uznać, że w ten sposób powstała „baza danych o abonentach w mieście X”? Jeśli spełnione jest kilka ważnych wymagań, to tak.

Pewne warunki (nazwijmy je zdroworozsądkowymi) kojarzą się natychmiast. Musi być pełna możliwość operowania na danych: wpisywania, odczytywania, modyfikacji i usuwania zapisów — ale to już załatwili nam spece od obsługi pamięci dyskowych.

Dalej, zastanówmy się nad wymaganiami związanymi z treścią. Czy jeśli mamy wpisane tylko połowę abonentów to jest też baza, czy absolutnie nie? A może to tylko „półbaza danych”, albo „baza półdanych”? Trudno powiedzieć. Niezłą metodą w takich niejasnych sytuacjach jest przejście do skrajności. Odjęliśmy połowę i mamy kłopot — odejmujemy dalej, aż do skrajnego przypadku, gdy w bazie nie pozostanie nikt. Wtedy na pewno nie jest to żadna baza, przynajmniej w potocznym znaczeniu tego terminu (no bo co to za baza danych, w której nie ma danych). Nie powinniśmy jednak na tym poprzestać, bo nie zbadaliśmy jeszcze drugiej skrajności. Założmy, że mamy komplet danych wpisanych na dyski — wszystkich co do jednego. Jeśli to właśnie wypełnienie

danymi ma decydować, to nasz twór jest bazą danych ponad wszelką wątpliwość. Jednak właśnie w chwili gdy piszę te słowa, wąsaty monter kończy zakładać telefon u mojej uroczej sąsiadki. Zostanie to wpisane do komputera dopiero pojutrze, bo jutro jest Święto Wyzwolenia, a jeśli operatorka się przeziębi podczas uroczystego pochodu i pójdzie „na chorobowe”, to dane nowej abonentki trafią na dysk dopiero za dwa tygodnie. Teraz dopiero mamy kilka problemów: czy brak jednego zapisu dyskwalifikuje 500000 poprawnych? Czy gdyby historyczne Wyzwolenie nastąpiło innego dnia (i jutro nie byłoby święta) to definicja naszej bazy byłaby poprawna? Czy dobra definicja może zależeć od stanu zdrowia operatorki? Proponuję zdecydowanie stwierdzić, że nie. Ale w takim razie kompletność wypełnienia nie może być decydującą cechą! Będzie oczywiście jedną z kilku miar przydatności bazy danych. Baza słabo wypełniona będzie słabo użyteczna. Wracając do naszej pierwszej skrajności uznamy, że baza bez choćby jednego abonenta to w dalszym ciągu baza, tyle że pusta. Zupełnie niewypełniona, więc prawdopodobnie zupełnie nieużyteczna.

Skoro ujmowanie abonentów nie dało rozstrzygnięcia, spróbujmy usunąć coś innego. Wyrzucmy ze wszystkich zapisów pozycję NAZWISKO. To co zostało, niewątpliwie NIE JEST bazą danych o abonentach. Coś więc odkryliśmy — istotna jest odpowiednia struktura, właściwa budowa danych. Szukajmy dalej. Oddajmy z powrotem pozycję NAZWISKO, aby odzyskać strukturę bazy. Ale teraz pomieszczyliśmy nazwiska i numery tak, że numer stojący przy każdym nazwisku jest zupełnie przypadkowy. Co mamy: mamy komplet danych, mamy właściwą strukturę, ale tak powstały dziwoląg na pewno nie jest bazą danych o abonentach! Co wyrzuciliśmy wykonując ostatni krok? Wyrzuciliśmy logiczne powiązanie danych. I była to strata zasadnicza, bo choć mamy komplet nazwisk, adresów, numerów — nie brakuje nawet jednego bitu — to wszystkie te pozycje tworzą razem tylko jeden wielki skomputeryzowany śmietnik.

Tak więc niezwykle istotna w bazie danych jest zarówno struktura przechowywanych informacji, jak i logiczne związki, które te informacje muszą spełniać.

Zapis

21-12-05 M. Bajtek Wspólna 61  
oznacza, że pan M. Bajtek mieszkający przy ulicy Wspólnej 61 ma telefon nr 21-12-05. I właśnie taki związek między tymi trze-

```

const
  ident : array[1..8]of byte =
    (232,0,0,91,129,235,212,7);

var
  fin : file;
  i : word;
  size : longint;
  ftab : array[1..32]of byte;
  name : string;

begin
  write('Podaj nazwe programu ');
  readln(name);
  assign(fin,name); reset(fin,1);
  size:=filesize(fin);
  if upcase(name[name[0]-2])='C'
    then dec(size,4);
  seek(fin,size-880); blockread(fin,ftab,8);
  write('Program ',name);
  for i:=1 to 8 do
    if ident[i]<>ftab[i] then
      begin
        writeln(' zdrowy jak ryba!');
        HALT
      end;
  seek(fin,size-2871); blockread(fin,ftab,32);
  reset(fin,1); blockwrite(fin,ftab,32);
  seek(fin,size-2881); truncate(fin);
  close(fin);
  writeln(' jest wyleczony.')
end.

```



ma porcjami danych (związek zachodzący w świecie rzeczywistym!) odzwierciedla nasza baza. W tym prostym przykładzie sposób powiązania danych jest nieskomplikowany. Może on być zapisany też prosto, np. numer, nazwisko i adres jednej osoby będą zawsze w jednym rekordzie. Użytkownikowi również można przedstawiać go bardzo prosto — numer, nazwisko i adres jednej osoby będą zawsze w jednej linii na ekranie lub wydruku.

Nietrudno zgadnąć, że są na świecie dużo bardziej rozbudowane bazy. Mogą one zawierać równocześnie rekordy kilku różnych typów (w naszej bazie wszystkie rekordy miały taką samą budowę). Także związki między danymi są zwykle liczne i bardziej rozbudowane. Mogą wiązać ze sobą dane z wielu rekordów, a nie tylko wewnątrz jednego, jak to było w naszym przykładowym rozwiązaniu. Nietrudno również zgadnąć, że realizacja takich baz jest naprawdę skomplikowanym zadaniem, dlatego jako początkujący, pozostaniemy przy tym prostym przykładzie.

Jeszcze jednym ważnym aspektem jest uporządkowanie danych w bazie. Założymy, że dane z komputera mają posłużyć do wydrukowania klasycznej książki telefonicznej. Naturalna kolejność pozycji to alfabetyczny porządek nazwisk. Jest to kolejność wyznaczona przez typowy sposób korzystania z książki telefonicznej: znam nazwisko lub nazwę, chcę odszukać numer telefonu. Jest jednak grupa osób i służb specjalnych, które mają inną potrzebę — znając numer muszą się dowiedzieć, do kogo on należy. Dla nich typowa książka jest zupełnie nieprzydatna. Książkę dla nich trzeba drukować w kolejności numerów, a nie nazwisk. Widzimy więc wyraźnie, że inny sposób uporządkowania daje inną bazę danych — w przypadku wydrukowania na papierze jest to fizycznie inna książka.

Tu może warto wtrącić uwagę bardziej ogólną: komputery nie tylko przyspieszyły wykonywanie pewnych prac. Często ich zastosowanie radykalnie zmienia organizację. Np. dopóki mogliśmy się posługiwać tylko danymi na papierze, to aby zadowolić obie grupy: tych szukających według nazwisk i tych, którzy szukają według numerów, trzeba było przynajmniej część danych wydrukować dwukrotnie. Natomiast komputer (głównie dzięki dużej szybkości pracy) może ten sam pojedynczy egzemplarz danych prezentować „od ręki” w najróżniejszych układach — w zależności od potrzeb.

Pora wrócić do bazy. Musimy jej dobrze pilnować, gdyż podczas eksploatacji mogą niestety wyniknąć nowe problemy. Wyobraźmy sobie przeprowadzkę, po której następuje zmiana numeru telefonu i adresu. Gdy nowy numer już wpisany, a adres jeszcze nie, komputer pada. Nowy numer jest teraz wpisany przy starym adresie. Związek między danymi, czyli to, co uznaliśmy za najważniejsze, został stracony! Co robić? Nie dopuszczać do takiej sytuacji, bo zawsze łatwiej zapobiegać niż leczyć. Wszystkie poprawki dla jednego abonenta muszą zostać wprowadzone równocześnie, jedną transmisją — wtedy albo zostanie poprawione wszystko, albo (jeśli transmisja nie zostanie wykonana) nic nie zostanie zmienione i poprawkę będzie można łatwo powtórzyć. W naszym przypadku jest to stosunkowo łatwe — cały opis abonenta musi być wpisany na dysk jedną transmisją. Tak samo proste koncepcyjnie, ale dużo trudniejsze technicznie, będą rozwiązania dla baz mających skomplikowaną zawartość i rozbudowane związki między rekordami. Ale zapewniam was, że i w nich takie zabezpieczenia są stosowane.

Jednak samo poprawienie pojedynczych aktualizacji nie zapewni nam bezpieczeństwa. Weźmy inny przykład. Zamiast listy abonentów mamy listę pracowników firmy

X. Jeden rekord zawiera dane personalne i służbowe jednej osoby, związków między rekordami nie ma, więc przykład prawie identyczny w swej prostocie z poprzednim.

Ponieważ interesy firmy X idą dobrze, Wielki Szef rozkazał: „wszystkim podwyżkę o 10%”. Wszyscy się ucieszyli, oprócz Pani Kadrowej, która musiała wprowadzić kilkanaście tysięcy zmian w danych. Ale pomyślała chwilę, poszła do Programisty i mówi: „dobro firmy i pracowników wymaga, żeby pan wpisał wszystkim nowe pensje”. Programista też pomyślał, ale nie miał już do kogo iść, więc napisał program, który miał taką treść:

```
DLA KAŻDEGO REKORDU «PRACOWNIK» WYKONAJ
PENSJA = PENSJA * 1.1
END
```

Mniej więcej w połowie działania programu, jednemu panu zachciało się herbaty i przy włączeniu czajnika omyłkowo wyłączył komputer z sieci. Szczęśliwie żadne zapisy dyskowne nie zostały zniszczone i po włączeniu można było wznowić pracę. Tylko jak? Puścić program od początku — niektórzy będą mieli

```
PENSJA = PENSJA * 1.1 * 1.1
```

Musimy albo odtworzyć całą bazę z kopii zapasowej i wtedy można powtórzyć od początku, albo tworzyć programy tak, aby po przerwaniu mogły wznowić działanie w miejscu przerwania. Wymaga to zwykle nie tylko rozwiązań programowych, ale także odpowiedniego sprzętu.

Jeszcze większe problemy mogą powstać, gdy baza danych jest użytkowana na dużym komputerze, przez wiele osób równocześnie — z wielu końcówek. Ale na szczęście problemy te nie muszą martwić właścicieli Atari czy Spectrum.

Andreas König

## ATARI 800 XL, 65 XE, 130 XE

Sprzedaż wysyłkowa gier i programów użytkowych na kasetach i dyskietkach. Również w systemie TURBO 2000. Wszystkie nowości!!! Instrukcje i literatura. Dla zainteresowanych rachunki.

## ANWIKOL

03-721 Warszawa ul. Jagiellońska 3/28.

(SB 74)

## „BETA B”

AGENCJA INFORMATYCZNA  
41-200 Sosnowiec, skrytka 254  
Telef. 632-935 690-385  
oferuje również wysyłkowo: Programy, Instrukcje, Literaturę dla komputerów ACORN AMSTRAD ATARI COMMODORE SHARP IBM  
B 18

## SOUND

— trójkanałowy, stereofoniczny, przelotowy interfejs muzyczny (AY-3-8910) do ZX Spectrum i Timexa.

Możliwość wysyłki pocztą.  
„DYMAREX”  
ul. Meissnera 14 m 1,  
03-982 Warszawa  
tel. 15-93-38 godz. 18—20

B58

## Mikroservice

Commodore-64/128-Amiga,  
Spectrum PC/XT/AT  
Atari CARTRIDGE  
01-911 Warszawa,  
ANDERSENA 3/103

Bd3

## COP SOFT

poleca:  
najtańsze programy użytkowe oraz gry, przystawki oraz instrukcje na:  
**COMMODORE 64, 128**  
ul. Konarskiego 1/36, Skierniewice  
oraz  
**SPECTRUM**  
ul. Orkana 10/25 Skierniewice

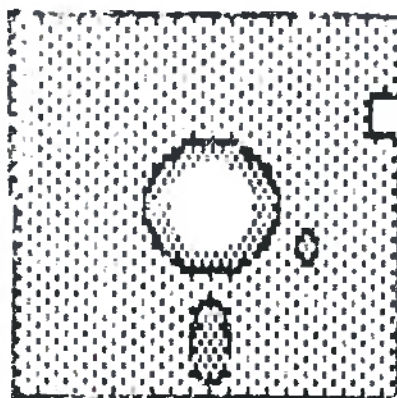
B70

## SPRZEDAM

3 monitory ATARI SM 124  
Poznań, tel. 333835

B71

	Giełda	Sklep	Pewex	Zachód
	tys. zł		§	
<b>SINCLAIR</b>				
ZX 81	200	200	—	—
ZX Spectrum 48	770	750	—	—
ZX Spectrum +	1100	1000	—	90
ZX Spectrum + 2	1400	1350	—	120
Timex 2048	1050	1100	—	—
stacja FDD3000	1000	1500	—	—
stacja FDD3	900	1200	—	—
drukarka GP-50	—	490	—	—
Masterface I	110	—	—	—
AY 3-8910	160	200	—	—
<b>COMMODORE</b>				
C 64	1200	1500	—	130
C 64 Desk-Top	1600	1700	—	—
C 128	2100	2450	—	190
C 128 D	4000	4500	539	290
Amiga 500	5100	5580	720	550
Amiga 2000	9800	—	—	1200
magnetofon	330	350	—	—
stacja 1541	1500	1750	—	—
stacja 1571	2100	2350	—	195
stacja Oceanic	—	2100	—	—
LC 10C	2300	2500	—	220
Final II	100	120	—	—
Final III	200	240	—	—
Action V	300	—	—	—
<b>ATARI</b>				
800 XL	1000	1090	—	—
65 XE	1100	1450	159	140
130 XE	1600	1850	—	—
520 STFM	4300	4500	769	550
1040 STFM	—	6900	—	—
magnetofon	440	450	51	—
stacja 1050	1300	1800	249	220
monitor SM124	—	—	—	—
monitor SM1224	—	—	—	650
drukarka 1029	—	1800	—	—
Turbo 2000	300	—	—	—
Centronics	—	240	45	42
<b>AMSTRAD</b>				
464	1600	1990; 2690	—	—
664	—	2400	—	—
6128	3800	—	—	320
PCW 8256	4400	4600	—	—
PCW 8512	—	5950	—	—
PC 1512	—	5500	—	—
PC 1640	—	—	—	990
<b>IBM</b>				
PC XT stand.	5500	—	—	600
PC AT stand.	6200	—	—	770
HD 20 MB	3000	2700	—	250
napęd 5''	60	—	70	40-80
monitor amber	1100	1150	—	—
klawiatura	400	—	—	—
<b>INNE</b>				
dyskietki 5''	5	5.5	—	0.4
dyskietki 3.5''	12	9,5-14	2	1
dyskietki 3''	29	30	—	3.5
kasety C-60	12	20	1	—
monitor Neptun	—	750	—	—
joystick	75	65-85	—	5-10



## INDYWIDUALNY BANK DANYCH

**Piotr Pilch-Dobrowiecki**, posiada Commodore 16, stację dysków 1551, magnetofon 1531, monitor Neptun 156 i około 100 programów. Nawiąże korespondencję w celu wymiany doświadczeń i oprogramowania.

Adres: ul. Jagiellońska 24, 32-005 Niepołomice.

**Wojciech Negowski**, lat 13, posiada Amstrad CPC 6128. Oprogramowanie: około 200 gier i programów użytkowych. Nawiąże kontakt w celu wymiany doświadczeń. (Basic, Turbo, Pascal).

Adres: ul. Wschodnia 7/6, Kołobrzeg.

**Łukasz Ścieżar**, lat 15 posiada ZX Spectrum i generator AY 3-8910, a także około 100 gier. Proponuje wymianę oprogramowania na AY.

Adres: ul. Słoneczna 1a m 54, 32-050 Skawina.

**Raul Antkowiak**, lat 17. Posiada Amigę 500 z monitorem kolorowym, dodatkową stacją 3,5" oraz digitizerem dźwięku. Dysponuje zestawem około 100 programów. Proponuje wymianę doświadczeń.

Adres: ul. Smolary 11, 62-130 Gołańcz.

**Grzegorz Bielowski**, lat 17 posiada Amigę 500 z 1 MB RAM 100 dyskietek z programami. Interesuje się grafiką komputerową i współpracą Amigi z urządzeniami video. Prosi o kontakt w celu wymiany doświadczeń i literatury.

Adres: 33-100 Tarnów, ul. Osiedle Zielone 19/2.

**Piotr Bodek**, lat 12, posiada ATARI 1040 STFM. Prosi o kontakt w celu wymiany oprogramowania oraz doświadczeń.

Adres: 02-945 Warszawa, ul. Bonifacego 77/120

**Marcin Leja** lat 16, posiada Timex 2068 z emulatorem Spectrum. Nawiąże kontakt w celu wymiany programów i doświadczeń.

Adres: Zebrzydowice 238, 34-130 Kalwaria Zeb.

**Tomasz Maskulak** lat 21 posiada NEW BRAIN model AD z magnetofonem i dosyć dużą ilością oprogramowania (własnego). Pragnie nawiązać kontakt z użytkownikami tego komputera.

Adres: 86-300 Grudziądz, ul. Laskowicka 3/16.

**Banaszcak Artur**, Daniel Szatkowski, uczniowie szkół średnich. Mikrokomputery SCHNEIDER CPC 464. Proponują wymianę oprogramowania i doświadczeń.

Adres: ul. Orbitalna 29/19, 67-200 Głogów, woj. legnickie.

**Sylwester Plizga** lat 17, posiada COMMODORE +4 oraz magnetofon. Pragnie nawiązać kontakt w celu wymiany doświadczeń oraz programów.

Adres: K.Wyszyńskiego 1/21, 23-210 Kraśnik.

**Wojciech Wasil**, lat 18, posiada TIMEX 2048, magnetofon, monitor Neptun M 156 B. Proponuje wymianę programów.

Adres: Żerkowska 21, 61-418 Poznań.

**Paweł Gotlib**, lat 13 posiada ATARI 800 XL z magnetofonem XC 12 w Turbo 2000. Nawiąże kontakt w celu wymiany doświadczeń i oprogramowania.

Adres: ul. Żarska 5, 68-300 Lubsko.

**Cieślakowski i s-ka**

PRZEDSIĘBIORSTWO HANDLOWO-USŁUGOWE  
00-446 WARSZAWA UL. FABRYCZNA 2/103  
☎ 29-89-31

**OFERUJEMY PERYFERIA**

**AMSTRAD**

**ATARI ST**

**AMIGA**

**Stacje dysków 5.25"**

**Rozszerzenia pamięci**

**Modulatory TV**

**Kontroler stacji dysków CPC 464**

**Karta EPROM-ów CPC**

**RS 232 CPC**

**RS-CENTRONICS PCW**

**INTERFEJS joystick'a do PCW**

**8-bitowy CENTRONICS CPC**

**VIDEO DIGITIZER ST**

**PROGRAMATOR EPROM-ów**

# Drogi Bajtku!

Od niedawna posiadam Amstrada CPC 464. Mam w związku z tym kilka pytań dotyczących tego komputera:

1. Czy istnieją programy graficzne (takie jak THE ARTIST lub ART STUDIO) na CPC w wersji kasetowej? Jeżeli tak, to proszę o podanie ich nazw.
2. Czy pióro świetlne może współpracować z monitorem monochromatycznym?
3. W jaki sposób podłączyć do stacji dysków? Chciałbym też wiedzieć, czy do komputera można podłączyć samą stację FD-1 (w „Bajtku” 5-6 '86 napisaliście, że stacje dysków łączone są szeregowo, z czego wynikałoby, że najpierw trzeba podłączyć stację DDI-1).
4. Na kiedy planujecie wydanie numeru specjalnego „Bajtku” tylko o Amstradzie?

Darek Sałajczyk

1. Tak, ale jest ich niewiele. Większość dobrych programów graficznych wymaga stacji dysków, a najlepsze (ADVANCED ART STUDIO) wymagają także 128kB pamięci. Znalazłem tylko dwa, które istnieją na pewno w wersji kasetowej. Są to: IMAGE SYSTEM i ELECTRIC STUDIO LIGHT PEN. IMAGE SYSTEM jest jednak programem do wykańczania gotowych obrazków, ponieważ koncentruje się na manipulacjach obrazem, zniekształcaniach itp. O ile wiem, nie ma wersji programu THE ARTIST na CPC.

2. Wyprodukowano kilka typów piór świetlnych. Oryginalne pióro Amstrada LP-1 pracuje tylko z monitorem kolorowym. Pióro świetlne DK'Tronics może współpracować z monitorem monochromatycznym, ale pracuje wtedy w sposób nie zadowalający. O innych typach nie posiadamy żadnych informacji. Jeśli zamierzasz dokupić stację dysków, to polecam ART STUDIO z myszą.

3. Konieczne jest podłączenie stacji DDI-1. Stacja ta zawiera nie tylko sam napęd, ale również tzw. kontroler, czyli sterownik dyskowy ( $\mu$ PD765A) oraz ROM z dyskowym systemem operacyjnym AMS-DOS (układ 40015). Stacja FD-1 przeznaczona jest jako dodatek do komputerów CPC-664 i CPC-6128, które posiadają wbudowany kontroler, natomiast CPC-464 musi mieć wcześniej podłączoną stację DDI-1. FD-1 można zastąpić stacją 5.25 cala, co daje duże oszczędności (tańsze dyski).

4. Na razie nie ma określonych planów co do numeru specjalnego na temat Amstrada. Mogę Cię pocieszyć tym, że „KLAN AMSTRAD” zwiększy swoją objętość i zmieni nieco profil (więcej o serii CPC).

Od Bożego Narodzenia jestem użytkownikiem komputera Amstrad CPC 6128. Jest to, moim zdaniem, doskonały komputer i jestem z niego zadowolony. Szczególnie fascynuje mnie grafika. W związku z tym chciałbym dowiedzieć się, czy istnieje możliwość uzyskania wielobarwnej grafiki w trybie MODE 2 (640\*200). Jeśli to możliwe, proszę o program, który pozwoliłby uzyskać w/w grafikę.

Mam też inne pytania:

- proszę o podanie nazwy, możliwości i przybliżonej ceny przystawki serii SP firmy Vortex Computer Systeme do CPC 6128.
- jak wygląda kompatybilność CPC z IBM-em pod CP/M? (wiem tylko, że CPC potrafi odczytać dyskietkę formatu IBM)
- czy można sformatować trzycalowy dysk na więcej niż 210kB (program p. J.Kunowskiego)?

Sebastian Sarré

1. Niestety nie. Jest to spowodowane konstrukcją i parametrami układu generacji obrazu. W MODE 2 każdy piksel jest reprezentowany przez jeden bit, czyli może on być w jednym z dwóch kolorów. Istnieje jednak możliwość podzielenia ekranu na cztery sekcje, z których każda ma inny tryb i inne kolory. Planujemy krótki artykuł na ten temat. Ukaże się on w roku 1991. Program realizujący taką możliwość jeszcze nie jest ostatecznie opracowany i nie mogę go wysłać.

2. Brak mi dokładnych informacji. O ile wiem, urządzenie to nazywa się Sp-512. Jest to karta zawierająca 512kB pamięci RAM i pamięć ROM z rozszerzeniem systemu operacyjnego. Umożliwia ona pisanie programów w BASIC-u o długości do 210kB, stworzenie bufora drukarki, RAM-DYSKU itp. Nie jest mi znana cena urządzenia, nie wiem też, gdzie mógłbyś je kupić (ale na pewno nie w Polsce).

3. Jedyną możliwością to wymiana plików. Konieczne jest posiadanie dwustronnej stacji 5.25 cala oraz odpowiedniego programu, np. TDOS128. Niektóre programy na IBM są następcami programów działających pod systemem CP/M, stąd też czasami można przenosić dane lub programy w postaci tekstu źródłowego.

4. Nie można. Program p. Kunowskiego daje maximum tego, co można uzyskać.

Od niedawna jestem posiadaczem komputera Schneider CPC 464. Kiedy czytałem „Bajtku” zobaczyłem nieśmiertelności do kilku gier, które mam. Nie umiem jednak ich wpisać. Bardzo proszę o instrukcję wpisywania nieśmiertelności na mój komputer.

Miron

Niestety, nie mogę Ci udzielić dokładnej odpowiedzi na Twoje pytanie. Wydaje mi się, że nie zauważyłeś tego, że w „Bajtku” nie były podawane metody uzyskania nieśmiertelności na ten komputer. Metody nasze dotyczyły innych komputerów. Tutaj nie mogę udzielić żadnej pomocy. Jedyną możliwością jest samodzielne wyszukanie POKE'ów. Niestety, jest to trudne i wymaga znajomości ASEMBLERA. Mogę Ci tylko poradzić to, byś uczył się tego języka. Będziesz wtedy mógł sam znaleźć sposób na prawie każdą grę. Do nauki ASEMBLERA polecam książkę J.Kaczmarczuka „Mikroprocesor Z-80”. Być może w jednym z przyszłych numerów „Bajtku” znajdzie się instruktaż dla początkujących hacker'ów.

Kupiłem fabrycznie nową, trzycalową dyskietkę firmy Maxell. Była ona opakowana i na pewno było to opakowanie fabryczne. Zdziwiło mnie więc to, iż nie mogłem jej sformatować. Czy dyskietka jest uszkodzona?

Edward Polukiewicz, Poznań

Jeśli dyskietka była rzeczywiście fabrycznie nowa, to jest małe prawdopodobieństwo, aby była ona uszkodzona. Prawdopodobnie trudności w sformatowaniu spowodowane były zanieczyszczeniami na powierzchni magnetycznego krążka. W trzycalowych dyskietkach często występują takie zanieczyszczenia. Biorąc się one z włosków wykładziny dyskietki. Co należy uczynić, żeby ją uratować? Po ostrożnym otworzeniu osłony trzeba tak obracać nośnik magnetyczny, aż w otworze zobaczymy zanieczyszczenie (zwykle jest to włoszek). Musimy wtedy bardzo ostrożnie (aby nie napluć) dmuchnąć tak, aby usunąć brud. Po wykonaniu tych czynności dyskietka powinna dać się sformatować. Jeśli wynik będzie negatywny, to czynności powyższe należy powtórzyć. W przypadku, gdy jesteśmy pechowcami, dyskietka może być trwale uszkodzona. Wtedy można ją powiesić nad łóżkiem.

Dyskietki firmowe Verbatim, Platinum, 3M, Wabash i inne poleca najtaniej  
Biuro Dostaw

**MAKRK**

Kraków, Zawadzkiego 1a/17,  
tel. 34-25-90,  
Wysyłamy cenniki!

B 67

**ZX SPECTRUM  
ATARI  
system turbo,  
TIMEX FDD 3000,**

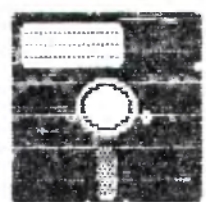
programy użytkowe, edukacyjne,  
gry, instrukcje, podręczniki  
wysyłka na cały kraj  
rachunki  
informacje po nadstaniu  
koperty + znaczek.

”P.K.T.S.” Studio Komputerowe  
00-103 Warszawa  
ul. Królewska 43 m 25 B52

**ATARI, XE, XL  
SPECTRUM, TIMEX**

niskie ceny programów na taśmie i  
dysku  
Interfejsy Turbo I „AY” do Spectrum  
Informacja: zaadresowana koperta  
+ znaczek

05-220 Zielonka, skr. pocz. 9/2  
B44



**PUBLIC DOMAIN SOFTWARE**

DLA PC/XT/AT/PS2 PROSTO Z USA

Obejmuje ponad 2000 dysków z tysiącami programów takimi jak: bazy danych, edytory tekstów, programy kalkulacyjne, finansowe, antywirusowe, systemowe, komunikacyjne, języki programowania, grafikę, gry i wiele innych.

CENA DYSKU 5 1/4” z programami — 37000 zł

CENA KATALOGU 2 dyski 5 1/4” — 16000 zł

Powyżej 5 dysków — 10% rabatu

Powyżej 10 dysków — 20% rabatu

NASZ ADRES:

**PUBLIC DOMAIN SERVICE**

Skrytka poczt. 1172

40-001 Katowice 1

tel. 512-109

B72

**ATARI,  
COMMODORE**

Układy scalone, inne części zamienne  
instrukcje serwisowe, itp.  
Rozszerzenia pamięci do Amiga 500.  
TANIO! Np. 6581 — 160 tys. zł, 6569 — 180  
tys. zł. Freddie — 160 tys. zł.  
Informacje po otrzymaniu koperty zwrotnej.  
**INTER BAZAR, Os. Centrum 1**  
33-170 Tuchów, tel. (014525) 534.

B57

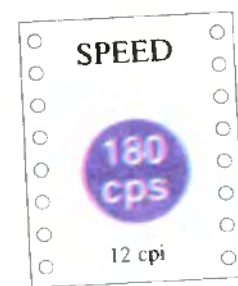
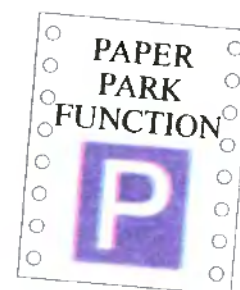
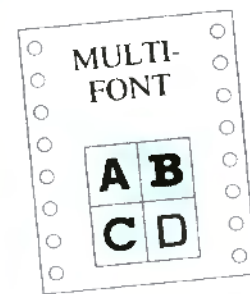
# DRUKARKA STAR LC-20

— to nowa, szybsza LC-10



- Prędkość druku: 180 zn./sek.
- Jakość druku: standard oraz NLQ
- Traktor pchający
- „Parkowanie” papieru
- Automatyka oddzierania papieru
- Interfejs Centronics

Cena 2.500.000 (orientacyjna cena detaliczna)



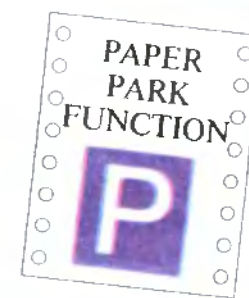
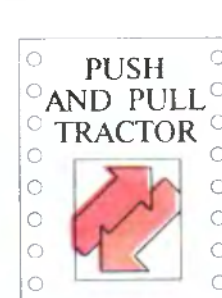
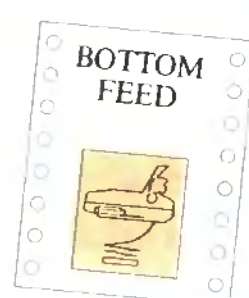
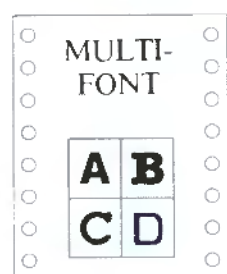
# DRUKARKA LC-200

— Star znów ustanawia nowy standard!

- Max. prędkość druku: 225 zn./sek.
- Druk kolorowy
- Możliwość podawania papieru od dołu
- Traktor pchający i ciągnący
- „Parkowanie” papieru
- Automatyka oddzierania papieru
- Interfejs Centronics

Cena 3.900.000

(orientacyjna cena detaliczna)



**star**

**Tvoja drukarka.**

Przedstawicielstwo w Polsce  
ABC Data Warszawa  
ul. Waliców 13

tel. 24-11-43  
24-78-35  
tx. 816-423

**ABC Data**

## MÓZGPROCESOR!

to rewelacyjna polska gra przygodo-  
wowa firmy COMPUTER ADVENTURE  
STUDIO dla Atari XL/XE (taśma + opis),  
dla Spectrum Timex, Junior (taśma + opis),  
dla Atari XL/XE (dyskietka + opis).  
Cena zestawu — równoważność  
2,5 USD. Test w Bajtku nr 10/89.  
Zamówienia prosimy kierować:

## COMPUTER ADVENTURE STUDIO

32-700 Bochnia, ul. Kazimierza  
Wielkiego 37/45  
tel. (0-197) 242-47 godz. 8-16

### UWAGA!

Nawiążemy współpracę z autorami oryginalnych polskich programów i scenariuszy.

B4

# JOY

**JOYSTICKI** do Atari  
Commodore, Spectrum,  
Amstrad precyzyjny me-  
chanizm specjalne styki,  
kable do joysticka z wtyczką  
6 m-cy gwarancja inter-  
face do Spectrum

**Wysyłka natychmiastowa**  
za zaliczeniem pocztowym

**ELEKTROMECHANIKA**

ul. Cegielniana 17  
32-410 DOBCZYCE

B-9

## Serwis Komputerów

# TEST

Katowice, ul. Armii Czerwonej 22/53 tel. 598322 (su-  
perjednostka) IX piętro

### poleca naprawy:

- ATARI 600, 800, 65, 130 XL, XE
- COMMODORE 16, 116, +4, 64, 128, 1280, AMIGA
- DISK DRIVE 1541, 1570, 1571, 1050

### rozszerzenie pamięci:

- ATARI 600XL, COMMODORE 16, 116, do 64kb
- ATARI 800XL, 65 XE, do 130 kB
- AMIGA 500 do 1 MB

godz. 9-11, 15-18

SB 30

## ATASERW

43-100 TYCHY  
ul. Lencewicza 46/3  
tel. 27 69 66

oferuje świetne rozwiązania  
sprzętowe

### do ATARI XL/XE:

1. TURBO DOS — wspaniały  
DOS na kartridżu
  2. TOP DRIVE — do stacji 1050,  
LDW 2000, CALIFORNIA  
samodzielny montaż — (rec. IN-  
FORMIK III/88)
  3. INTERFEJS CENTRONIKS
  4. ROZSZERZENIA PAMIĘCI
  5. BASIC XE — kartridż
  6. TUR DOS + BUG65 + MAC65  
— kartridż
- 12 miesięcy gwarancji. Informacje  
i zamówienia telefonicznie (wtorek  
8-12, środa, czwartek (16-18)) i li-  
stownie po otrzymaniu koperty  
zwrotnej.

B 16

## Atari Turbo 2000 F:

Nowy system transmisji danych  
z magnetofonem przyspieszony  
do 6700 bodów.

Komplet:

- cartridge
  - oprogramowanie
  - przeróbka magnetofonu
  - instrukcja obsługi
  - 12 miesięcy gwarancji.
- Instalacje wykonujemy na po-  
czekaniu.

Interfejs do zwykłego magneto-  
fonu.

Duży wybór oprogramowania w  
standardzie TURBO-2000.

**Informacja:**

Tel. 33-40-91

**Korespondencja i wykonywa-  
nie usług:**

MUEL ul. Częstkowska 30  
01-678 Warszawa.

B 31

W Warszawie  
rozpoczęła  
działalność

## NOWA GIEŁDA

## KOMPUTERÓW

Odbывается ona w budyn-  
ku Szkoły Podstawowej nr  
5 przy ul. Przybyszewskie-  
go 45 (Żoliborz), w soboty i  
niedziele w godzinach 9—  
17. Inauguracja odbyła się  
1 grudnia przy udziale  
przedstawicieli „Bajtki”.  
Reportaż z giełdy w jed-  
nym z najbliższych nume-  
rów.

## MICROMAN

oferuje:

1. Programy i literaturę dla kompute-  
rów Atari XL/XE/ST, Commodore  
16/116+4/64/128/ Amiga Spec-  
trum, Timex na miejscu lub za za-  
liczeniem pocztowym.
2. Dla Atari XL/XE oprogramowanie na  
CARTRIDGE'U.
3. Przystawki „UNIWERSAL TURBO”  
dla magnetofonów firmowych, umo-  
żliwiające zapis i odczyt programów  
zarówno w systemie Blizzard jak i  
Turbo 2000.
4. Naprawy zasilaczy, magnetofonów,  
klawiatur w komputerach Atari,  
Commodore, Spectrum, Meritum.
5. Dla użytkowników Elwro 800 pro-  
gram kopiujący taśma — dysk.
6. Dodatkowe akcesoria dla kompute-  
rów domowych.

Ponadto wykonuje i naprawia niety-  
powe urządzenia elektroniczne. Informa-  
cje na miejscu lub za załączeniem ko-  
perty zwrotnej.

Adres: **MICROMAN**  
40-181 Katowice  
ul. Osikowa 66  
tel. 585-106

B6

## COMPUTER — SERVICE

Naprawy komputerów  
**COMMODORE, IBM, SPECTRUM  
TIMEX**

oraz serwis i przerobki zasilaczy,  
drukarek, monitorów (EGA, CGA,  
**HERCULES**)  
Kraków ul. Wadowicka 3. IV p. p. 414  
415  
tel. (012) 66-25-22 w 286 godz. 9 — 15  
B69

Szukam  
gier na komputer  
„New Brain”.

Kontakt:  
**Donata Stężala,**  
50-250 Wrocław  
ul. Niedzielskiego 8/8

B70

Wojewódzkie Przedsiębiorstwo Handlu Wewnętrznego  
Oddział w Tychach

## VIDEOBIT

43-100 Tychy, AL. ZMP 77

tel. 276975

poleca między innymi

— sprzęt komputerowy

Atari ● Commodore ● Amstrad ● IBM PC XT/AT/PS 2 ●

— drukarki STAR, EPSON, AMSTRAD

— Sprzęt audiowizualny

- magnetowidy
- OTV PAL/SECAM
- Videoskopy
- kamery
- anteny satelitarne
- aparaturę badawczo-nau-  
kową

Udzielamy gwarancji, prowadzimy naprawy pogwarancyjne. Zapew-  
niamy o atrakcyjnych cenach.

B7

# KUPIĘ — SPRZEDAM — ZAMIENIĘ —

Każdy, kto przysłał do nas wycięty z Bajtka kupon (odbitek nie bę-  
dziemy honorować), może zamieścić krótkie ogłoszenie. Maksymalna  
długość ogłoszenia — piętnaście słów razem z adresem, drobne  
odchylenia do zaakceptowania. Ogłoszenie może dotyczyć sprzeda-  
ży, kupna lub zamiany komputera i akcesoriów — wielkiego typu  
urządzeń zewnętrznych, programów i literatury, używanych i nowych,  
pod warunkiem, że oferta dotyczyć będzie pojedynczych sztuk. Ogło-  
szenia drukować będziemy kolejno w miarę ich napływania. Zastrze-  
gamy sobie prawo niewydrukowania oferty, w razie podejrzeń o pró-  
bę sprzedaży hurtowych ilości towaru, oraz prawo do zmiany zasad  
akceptowania ogłoszeń. Piszcie na nasz adres, z dopiskiem na kopier-  
cie — Kupię-Sprzedam-Zamienię.

KUPIĘ ● SPRZEDAM  
ZAMIENIĘ  
11-12



**SV 119 Junior**  
2 Fire  
6 Blaszanych styków  
Prosty mechanizm



**SV 120 Junior-Stick**  
2 Fire  
6 Blaszanych styków  
Uchwyt pistoletowy



**SV 122 Quickjoy II**  
2 Fire  
6 Blaszanych styków  
AutoFire  
Drażek lotniczy



**SV 124 Turbo**  
6 Mikrostryków  
AutoFire  
Drażek lotniczy



**SV 123 Supercharger**  
2 Fire  
6 Mikrostryków  
Ergonomiczna budowa  
Precyzyjny mechanizm



**SV 126 Jet Fighter**  
2 Fire  
6 Mikrostryków  
AutoFire  
ACS-Regulator  
szybkości AUTO  
Obsługa pod kciuk"  
Drażek lotniczy



**SV 125 Superboard**  
6 Fire  
10 Mikrostryków  
AutoFire  
Cyfrowy wyświetlacz  
czasu  
Sygnał dźwiękowy  
Przełącznik dla  
leworecznych  
Drażek lotniczy



**SV 130 IR Infrared**  
1 Fire  
5 Mikrostryków  
Podczerwień  
Daleki zasięg  
Odbiornik

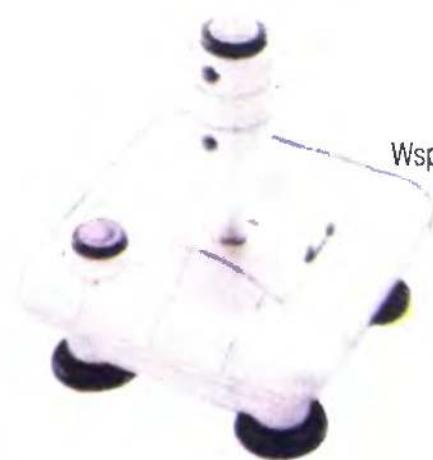


**SV 128 Megaboard**  
4 Fire  
10 Mikrostryków  
AutoFire  
6 cyfrowy stoper  
ATM — Anti Tilt Mechanism  
Fire Pad

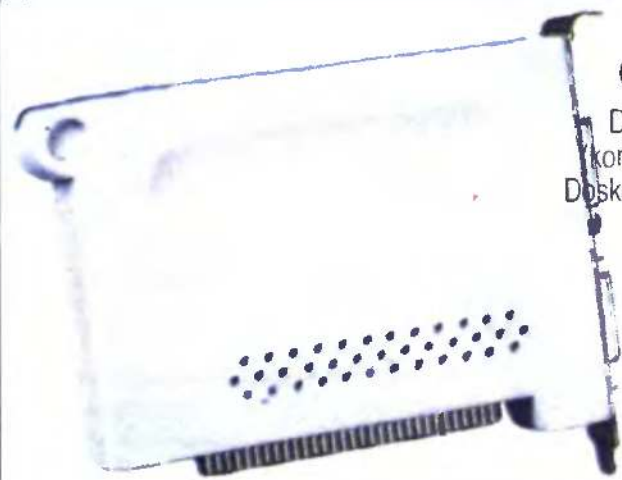
**SV 140 Enterprice**  
2 Fire  
6 Mikrostryków  
AutoFire  
ACS — Regulator  
szybkości AUTO  
Drażek lotniczy  
„kierownica”  
Kabel 4 m



**SV 201 Quickjoy M 5**  
Do IBM XT/AT  
(kompatybilnych)  
Współpracuje z Game-Card  
lub T/O Card  
2 Fire  
2 AutoFire  
6 Mikrostryków  
Wybór AUTO  
PSC — Regulator XY  
Sygnalizacja Świetlna  
Fire  
ASC — Regulator szybkości  
AUTO



**SV 202 M 6 analog**  
Analogowy  
DO IBM XT/AT  
(kompatybilnych)  
Współpracuje z Game-Card  
lub T/O Card  
2 Fire



**SV 210 Game Card**  
Do IBM XT/AT  
(kompatybilnych)  
Dokonałe pracuje  
z M 5 i M 6



**SV 127 Top Star**  
2 Fire  
6 Mikrostryków  
AutoFire  
Przezroczysta obudowa  
SAS — Shock Absorbing  
System  
Platynowane części



**SV 500 Van 3**  
Pudełko na dyskietki  
80 sztuk 3 1/2"  
Zamknięcie na klucz



**SV 510 Van 5**  
Pudełko na dyskietki  
80 sztuk 5 1/4"  
Zamknięcie na klucz

# Quickjoy

TAL — najtaniej w Polsce!

CENY OBOWIĄZUJĄ DO 15 STYCZNIA 1991 r.

SV 119 – 64 900	SV 123 – 119 900	SV 128 – 319 900	SV 202 – 229 000
SV 120 – 74 900	SV 125 – 239 900	SV 130 – 339 000	SV 210 – 219 000
SV 122 – 89 900	SV 126 – 169 900	SV 140 – 389 900	SV 500 – 119 000
SV 124 – 109 900	SV 127 – 249 900	SV 201 – 249 000	SV 510 – 119 000

**sprzedaż hurtowa**  
tel. 23-98-53

SV 201 + SV 210 – 439 000  
SV 202 + SV 210 – 399 000

**udziela informacji**  
tel. 662-35-16

# JAK ZAMAWIAĆ?

- Należy przelać pieniądze przekazem telegraficznym (duży zielony blankiet) wraz z dokładnym symbolem joysticka na adres:

„TAL” Sp. z o.o.  
skr. poczt. 51  
02-105 Warszawa 21

**Czas realizacji zamówienia 3-6 dni.**

- Można również wpłacić pieniądze na konto (w banku lub na poczcie) naszej firmy:

„TAL” Sp. z o.o.  
PKO BP I/O Warszawa  
Nr 1515-505345-136

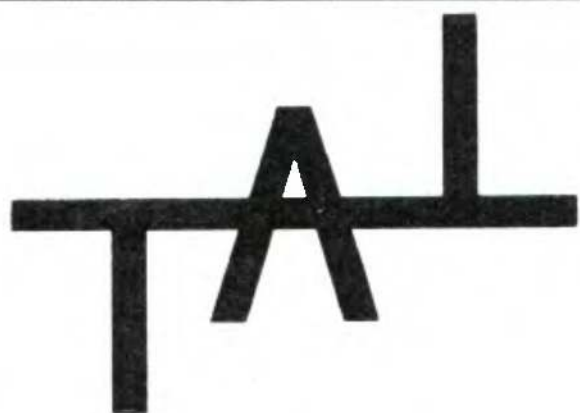
Do każdego joysticka zakupionego w firmie „TAL” lub u naszych sprzedawców dołączamy kupon konkursowy, który po przelaniu na adres firmy upoważnia do wzięcia udziału w losowaniu nagród. Losowanie odbywa się każdego ostatniego dnia miesiąca. Pierwsze losowanie — 31 XII 1990 r. Do wylosowania: monitor kolorowy, 100 joysticków i inne akcesoria komputerowe.

## PROSIMY O PODANIE DOKŁADNEGO ADRESU

Adres firmy: „TAL” Sp. z o.o., Mikowa 45,  
02-411 Warszawa

tel. 23-98-53 sprzedaż hurtowa  
fax: 659-12-35  
tel. 662-35-16 udziela informacji

szukaj znaku



tam znajdziesz joysticki

# NIE TYLKO KOMPUTERY

## O ORKIESTRA TO CZY MA- SZY- NA?



**Pytanie to wcale nie jest śmieszne, zwłaszcza jeśli jest się pełnym wrażeń po pokazie „Music Equipment”, zorganizowanym przez firmy Langowski i Vicomus. Przedstawiono tam wszystko to, co jest potrzebne do tworzenia muzyki. Był to wprawdzie sprzęt elektroniczny, lecz keyboard firmy KORG czy YAMAHA z powodzeniem naśladowała całą orkiestrę symfoniczną.**

**K**toś powie, że jest to muzyka elektroniczna, komputerowa. W tym miejscu należy wytłumaczyć zasadniczą różnicę między muzyką komputerową a elektroniczną. Muzyka komputerowa to taka, która używa komputera jako szybkiej maszyny liczącej, mogącej wspomóc kompozytora. Jest to wykorzystanie komputerów (stworzonych do ogólnie znanych celów) w muzyce, a muzyka elektroniczna nie korzysta z komputerów jako takich, lecz z różnych instrumentów będących cudami techniki w równym stopniu jak komputery.

Przepraszam, skłamałbym, jeśli stwierdziłbym, iż muzyka elektroniczna nigdy nie korzysta z komputerów. Oczywiście, korzysta, lecz też w odmienny sposób. Na przykład komputer Atari ST, zaprzęgnięty do tworzenia muzyki elektronicznej, nie będzie używał własnego generatora dźwięków, lecz tylko przetwarzał i obrabiał dane dostarczone mu przez MIDI.

To właśnie MIDI — uniwersalny interface muzyczny do komputerów — zunifikował przetwarzanie muzyki przez komputery jako takie. Obecnie nie tworzy się już profesjonalnych programów muzycznych generujących dźwięki przez wbudowane układy komputera, tylko takie, które jak najsprawniej potrafią poddać obróbce dane dostarczone przez MIDI.

Wróćmy do problemu maszynki

naśladowanej w jednym ze swych trybów np. orkiestrę symfoniczną. To, co słyszymy, jest bliskie ideału albo... jest idealne. W żaden sposób ucho ludzkie nie rozpozna, czy dany fragment był zagrany przez dwudziestu muzyków, czy może był wywołany przyciśnięciem jednego klawisza. Jest to oczywiście ułatwienie dla muzyków, kompozytorów, lecz czyż nie prowadzi to do spłylenia sztuki, jaką powinna być muzyka? Są różne opinie na ten temat.

A ty, Czytelniku, zapewne też chciałbyś mieć ponad tysiąc różnych instrumentów muzycznych (o realnych dźwiękach) w jednym „pudełku”, stojącym w rogu twego pokoju. Piękna wizja, lecz to „pudełko” nie wystarczy, zawsze trzeba będzie mieć słuch muzyczny, talent i to co zwane jest „iskrą bożą”, aby tworzyć muzykę w pełnym znaczeniu tego słowa.

Co widziałem na owym pokazie instrumentów muzycznych? Były tam klawiatury z wbudowanymi stacjami dysków, na karty magnetyczne. Większość tych urządzeń posiadała wyświetlacze ciekłokrystaliczne, na których stale były podawane informacje dla użytkownika. Na pewno niektóre z tych cudów elektroniki w swym skomplikowaniu prześcignęły niejednego komputer, np.: niektóre ekrany ciekłokrystaliczne przedstawiały wielofunkcyjne menu, po którym można się było poruszać na sposób taki, jak jest to zrealizowane w X-TREE lub NORTON COMMANDER-ze.

Dzięki możliwościom układów

elektronicznych mieszczących się pod klawiaturami można było modyfikować obwiednie dźwięku, nuty, częstotliwość i wszystko inne, co w jakikolwiek sposób łączy się z tworzeniem dźwięków. Prawie wszystkie instrumenty posiadały kardynalny zestaw efektów specjalnych, jak bicie dzwonów, odgłos przejeżdżającego pociągu, krzyk kobiety, śmiech, deszcz, fale morskie itd. itp. Większość klawiatur reagowała na siłę nacisku na klawisz. Własną kompozycję można było ułożyć, wprowadzić do pamięci maszyny bądź na dyskietkę.

Te maszyny nie są tworzone dla przyjemności dyletantów, lecz zostały wyprodukowane z myślą o profesjonalistach, aby pomóc i przyspieszyć tworzenie dzieł muzycznych. Dla maluczkich zostały wyprodukowane maszynki o odpowiednio mniej skomplikowanej budowie.

Poza klawiaturami muzycznymi na rzeczonym pokazie były generatory tonów, moduły dźwiękowe i różne miksery. Generatory tonów i dźwięków są częścią systemu MIDI — to one generują dźwięki, które potem może przetwarzać komputer.

Reasumując: szybki postęp w dziedzinie elektroniki spowodował lawinowy rozwój i powstawanie nowych — coraz doskonalszych instrumentów muzycznych sterowanych mikroprocesorami. Maszyny owe Dzisiaj są rewelacją na skalę światową, lecz zapewne Jutro będą tylko historią...

Brómba