

2

MIKROKOMPUTEREM NA TY

# Bajtek

NR INDEKSU 353965  
PL ISSN 0860-1674

MIESIĘCZNY DODATEK DO SZTANDARU MŁODYCH

NR 2(26)

LUTY

1988

CENA 100 ZŁ

AMSTRAD INFO

MUZYKA WŁASNA, NUTY KOMPUTERA

SM  
SZTANDAR MŁODYCH

DMP3160

MIGAJĄCY KURSOR

BUFORY

INWAZJA WIDEOFONÓW  
SABOTEUR II

# NA GORĄCO

Prawie równocześnie z ukazaniem się tego, lutowego numeru naszego pisma, znajdzie się też na rynku numer specjalny „Bajtka” poświęcony wyłącznie komputerom Atari. „BAJTEK-ATARI” jest odpowiedzią na prośby naszych Czytelników, a jednocześnie zapoczątkowuje całą serię dodatków specjalnych przeznaczonych dla użytkowników poszczególnych typów komputerów osobistych dostępnych na naszym rynku.

Od dawna otrzymywaliśmy listy, których nadawcy mieli do nas pretensje o to, że kolumn klanowych, poświęconych temu właśnie komputerowi, który oni mają, jest za mało. Świadczyło to częściowo o nieważnym czytaniu innych rubryk, dotyczących spraw wspólnych dla wszystkich typów komputerów, ale z drugiej strony stanowiło też dla nas wyraźny sygnał, że potrzebne jest na rynku większe różnicowanie periodycznych wydawnictw komputerowych. W swoim czasie zjawisko takie miało już miejsce na zachodnich rynkach pism komputerowych, nie ma więc nic dziwnego w tym, że powtórzyło się również u nas. Tam pojawiły się w odpowiedzi na to zapotrzebowanie pisma typu „Amstrad-User” czy „Spectrum-User” — u nas, mamy nadzieję, takim odpowiednikiem będą monotematyczne wydania dodatkowe „Bajtka” adresowane do użytkowników komputerów Atari, Commodore, Amstrada i Spectrum.

Wkrótce po „ATARI-BAJTKU” powinien pojawić się w kioskach Ruchu „BAJTEK-COMMODORE”, następnie kolejny „BAJTEK-ATARI” (jako że ten typ komputera zdominował nasz rynek), a potem przyjdzie kolej na inne komputery. Ponieważ są to wydania dodatkowe, więc nie obejmuje ich prenumerata — przypominam o tym, aby nie było nieporozumień.

Oczywiście, to swoiste rozrośnięcie się i usamodzielnienie kolumn klanowych nie oznacza wcale, że zrezygnujemy z nich na łamach „Bajtka”. Mają tu one swoje stałe miejsce — z tym że będziemy starali się prezentować w nich materiały najatrakcyjniejsze, mogące zainteresować również osoby dysponujące innymi komputerami. Będziemy zresztą starali się specjalnie tę uniwersalność materiałów zamieszczanych na kolumnach klanowych podkreślać i wskazywać.

A teraz kilka uwag na gorąco z trwających właśnie targów — wystawy „Komputer 88”. To co się dzieje w momencie pisania tych słów na pierwszych czterech piętach warszawskiego Pałacu Kultury i Nauki zaskoczyło wszystkich. Impreza, która odbywa się w Polsce zaledwie po raz trzeci już stała się instytucją tak znaczącą, że trudno sobie nawet wyobrazić bez niej lotowy pejzaż stolicy. Wykorzystana do ostatniego metra kwadratowego powierzchnia targowa, zaangażowane twarze oferentów i potencjalnych nabyw-

ców, tłumy „gości zaproszonych” od chwili otwarcia do zamknięcia wystawy — to najlepiej świadczy o sukcesie.

Dzięki strategicznym decyzjom podjętym w sztabie ludzi odpowiedzialnych za reformę gospodarczą powstał błyskawicznie w Polsce rynek komputerowy z prawdziwego zdarzenia. Powtarzamy — nie wbrew reformie, jak niektórzy lubią narzekać, tylko właśnie w wyniku i w duchu reformy! Dzięki temu dziesiątki, a w tej chwili już nawet setki małych i średnich firm komputerowych mogło na serio, z nadzieją na niezłe zyski, zająć się informatyzacją poszczególnych dziedzin naszej gospodarki. Warszawska wystawa była właśnie przeglądem najprężniejszych w tej dziedzinie. Trzeba przyznać, że był to przegląd imponujący.

Okazało się przy okazji, że wykształcił się na styku sektora zagraniczno-prywatnego z państwowym i spółdzielczym specyficzny typ polskiego menadżera branży komputerowej. Zdominowali warszawską imprezę młodzi ludzie między 20 a 30 rokiem życia. Dobrze władający w większości językiem angielskim, otwarci w kontaktach, precyzyjni w kalkulacjach a jednocześnie umiejący szybko podejmować decyzje. W nich nasza nadzieja!

Szkoda tylko, że tak trudno było się dostać na „Komputer 88” ich młodszymi kolegami — uczniom szkół podstawowych i średnich. Do godziny 16.00 impreza była przeznaczona tylko dla „gości zaproszonych”, dopiero między 16.00 a 18.00 mogli przez trzy dni obejrzeć ją „normalni” ludzie. Te sześć godzin — i to w nienajlepszej porze, bo dawno po zajęciach szkolnych — to było wszystko, na co mogli liczyć warszawscy uczniowie. Wycieczki z innych miast były zupełnie bez szans!

Można zrozumieć motywacje organizatorów i wystawców, chcących przede wszystkim „robić biznes”, ale trudno zgodzić się na traktowanie po macoszem potrzeb tych, którzy do „biznesu” będą jeszcze przez kilka lat dorastać. Największa wystawa komputerowa w Polsce powinna również w szerszym stopniu spełniać funkcje edukacyjne, powinna stwarzać młodzieży możliwość choćby „napatrzenia się” na najnowszą technikę komputerową. Takie są potrzeby społeczne i organizatorzy nie powinni ich w żadnym razie lekceważyć! Sześć godzin dla szerokiej publiczności to stanowczo za mało. Redakcja „Bajtka” wyraża nadzieję, że w programie przyszłorocznej imprezy w szerszym stopniu uwzględnione zostaną interesy szerokich rzesz młodzieży!

Tyle impresji na gorąco. Pełna relacja z „Komputera 88” — już w następnym numerze „Bajtka”.

Waldemar Siwiński

## SZANOWNY PANIE REDAKTORZE!

Po publikacji na temat Warsaw Basic w Bajtku nr 7 z tego roku utwierdziło się moje przekonanie, że sprawa edukacji naszej młodzieży w zakresie informatyki i innych przedmiotów w oparciu o mikrokomputery nie może być dalej zaniedbywana, leżąc ciągle w sferze planów i badań bez realizacji, tym bardziej, że obecnie już posiadamy wszystko, co jest potrzebne do takiego przedsięwzięcia, a więc polski interpreter-program Warsaw Basic, czekające moce produkcyjne do jego produkcji i pewną ilość komputerów w szkołach i na wyższych uczelniach.

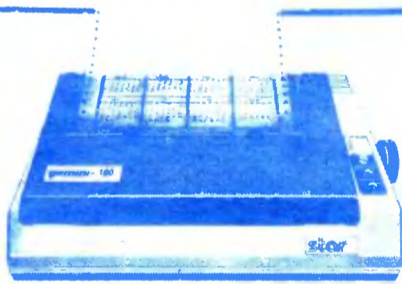
Szkoły nasze obecnie są w nikłym procencie wyposażone w jakiegokolwiek komputery. W tej nikłej ilości występują takie typy jak Commodore, Atari, Spectrum Amstrad i polski Meritum. Polski Junior 800 dopiero będzie wprowadzony.

Taka różnorodność i niedostatek uwarunkowane są sytuacją ekonomiczną i rynkową naszego kraju jest jak mi się wydaje w pewnym sensie powodem opóźnienia w zasadniczych decyzjach i realizacjach w zakresie edukacji informatycznej i innej w oparciu o komputery naszej młodzieży.

Ta różnorodność i wypływające z tego niezdecydowanie oraz żywiołowe edukowanie się naszej młodzieży nie jest najlepszym rozwiązaniem. Można przecież w tej sytuacji zaproponować strefowe oparte na systemie wojewódzkim wyposażanie szkolnictwa w dostępne różne komputery.

Natomiast w drugim etapie po okrzepnięciu naszego przemysłu mikrokomputerowego, zdobyciu doświadczenia w szkołach i zużyciu się obcych maszyn cyfrowych wprowadzić generalnie nasz np. Junior 800 standardowo wyposażony w Warsaw Basic. Zresztą ten system wyposażania komputerów jednocześnie przy sprzedaży w programy jest powszechnie stosowany.

W historii innych dziedzin naszego życia ekonomicznego wiele mieliśmy przykładów podobnych rozwiązań, jak zaproponowane po-



wyżej. W obecnej naszej sytuacji pozwoli to na szybkie skasowanie opóźnienia edukacyjnego naszej młodzieży w tym zakresie.

W oparciu o powyższe kompletnym nieporozumieniem byłoby gdybyśmy tylko w oparciu o obce i obcojęzyczne programy wychowali naszą młodzież tym bardziej, że swój i to jeden z najlepszych interpreterów-programów WSO Warsaw Basic piszący po polsku i rosyjsku mamy. Został on opracowany przez dr Krzysztofa Gajewskiego i doc. dr hab. Bogusława Radziszewskiego z IPPT PAN w Warszawie, w ramach pracy nad Warszawskim Systemem Operacyjnym. Ten interpreter-program został zbudowany na najużyteczniejszej i najpopularniejszej na świecie strukturze Basic tak jak kiedyś nasz język na strukturze gramatyki łacińskiej.

Elementy edukacyjne W.B. w zakresie szkolnej informatyki są następujące:

1. Nauka programowa i jego zaawansowanych elementów na bazie Basic (p. Bajtek) biblioteka podprogramów, programowanie proceduralne i strukturalne.
  2. Geometria, obrazy i figury, płaskie przestrzenne i ich obroty.
  3. Język polski na ekranie i na papierze.
  4. Praca na komputer w dowolnym języku (ekran papier) w tym wydaje się być bardzo interesującą propozycją pracy w języku rosyjskim, szczególnie w początkowych klasach nauczania.
  5. W.B. daje możliwość efektywnej nauki fizyki, chemii, matematyki, historii i innych przedmiotów w języku polskim.
- Ten bezsprzecznie jeden z najlepszych na świecie interpreterów jak uznaje to opinia środowiska informatyków powinien być jak najszybciej wprowadzony na wyposażenie szkolnictwa. Na razie naturalnie do szkół posiadających mikrokomputery typu Commodore, gdyż na ten typ komputera został on w pierwszym rzędzie opracowany. Po zapadnięciu długo oczekiwanych, co do wyposażenia szkolnictwa w ten język — decyzji może on być szybko dostosowany do każdego innego komputera w tym także istniejących w Wyższym Szkolnictwie IBM-ów.

W sumie wprowadzenie WSO i WB do naszego systemu edukacyjnego niższego i średniego szczebla a po przystosowaniu WB do IBM-ów także do Szkolnictwa Wyższego wydaje się być już palącą koniecznością a nie jedną z alternatyw.

Dr inż. Krzysztof August Kunert  
Warszawa

## WYBIERZ SAM

GRA O JUTRO	
Muzyka własna, nuty komputera . . .	3
Komputerze — wszystkiego najlepszego . . . . .	4
PROGRAMOWAĆ MOŻE KAŻDY	
Zmienne dynamiczne . . . . .	5
KLAN ATARI	
Porty joysticków (1) . . . . .	7
Tajemnice Atari (6) . . . . .	7
Migający kursor . . . . .	8
Atak! . . . . .	9
KLAN COMMODORE	
Polskie znaki . . . . .	10
Jak rozbudować interpreter? . . . . .	11
Papierowy IBM . . . . .	11
Czwarte wcielenie dinozaura . . . . .	12
KLAN SPECTRUM	
Zepsułem program . . . . .	13
Od środka (2) . . . . .	14
KLAN AMSTRAD/SCHNEIDER	
Zamieszanie na ekranie . . . . .	20
INFO . . . . .	21
CO JEST GRANE	
Saboteur II . . . . .	16
Kung-Fu Master . . . . .	18
Barbarian . . . . .	19
NASTĘPNY KROK	
System operacyjny . . . . .	22
Bufory . . . . .	23
TEST	
DMP 3160 . . . . .	24
JAK TO ROBIĄ INNI	
Bagażowy . . . . .	25
TYLKO DLA PRZEDSZKOLAKÓW	
Wieża i kopnięty kwadrat . . . . .	31
NIE TYLKO KOMPUTERY	
Inwazja wideofonów	

### „BAJTEK” — MIESIĘCZNY DODATEK DO „SZTANDARU MŁODYCH”

ADRES: 00-687 Warszawa, ul. Wspólna 61. Tel. 21-12-05  
Przewodniczący Rady Redakcyjnej: Jerzy Domański-  
redaktor naczelny „Sztandaru Młodych”.

**ZESPÓŁ REDAKCYJNY:** Waldemar Siwiński (z-ca redaktora naczelnego „SM” — kierownik zespołu „Bajtka”), Roman Poznański (z-ca sekretarza redakcji „SM” — sekretarz zespołu „Bajtka”), Krzysztof Czernek, Sławomir Gajda (red. techniczny), Andrzej Gogolewski, Andrzej Kowalewski, Andrzej Podulka, Sławomir Polak, Wanda Roszkowska (opr. graficzne), Kazimierz Treger, Marcin Wałigórski, Roman Wojciechowski. Zdjęcia w numerze: Leopold Dzikowski.

**Klasy redagują:**  
Commodore — Klaudiusz Dybowski,  
Amstrad-Schneider — Tomasz Pyć, Sergiusz Wolicki,  
Spectrum — Marcin Przasnyski, Michał Szuniewicz,  
Atari — Wiesław Migut, Wojciech Zientara.

Fotoskład — Tadeusz Olczak,  
Montaż offsetowy — Grażyna Ostaszewska,  
Korekta — Maria Krajewska, Zofia Wóltańska  
**WYDAWCA:** RSW „Prasa-Książka-Ruch” Młodzieżowa  
Agencja Wydawnicza, al. Stanów Zjednoczonych 53,  
04-028 Warszawa. Telefony: Centrala 13-20-40 do 49,  
Redakcja Reklamy 13-20-40 do 49 w. 403, 414.  
Cena 100 zł.  
Skład technika CRT-200, przygotowalnia offsetowa i druk:  
PRASOWE ZAKŁADY GRAFICZNE RSW „PRASA-  
KSIĄŻKA-RUCH” w Ciechanowie, ul. Sienkiewicza 51.  
Nr zlecenia 010188 n. 150.000 egz. U-113



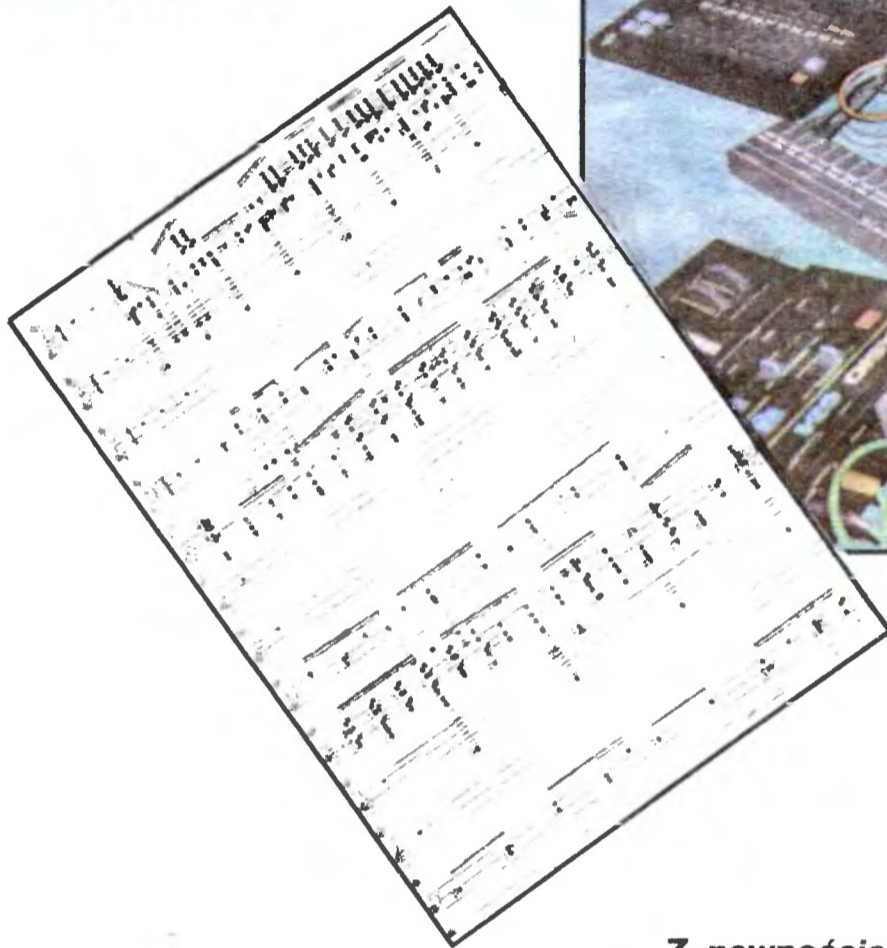
Bajtek

# MUZYKA WŁASNA NUTY KOMPUTERA

## Rozmowa z kompozytorem i muzykiem — Krzysztofem Sadowskim



Często kiedy patrzę na zapis nutowy swojej kompozycji, jestem zdumiony...



— **Grasz na komputerze?**

— Tak

— **Od kiedy?**

— Od paru miesięcy. Instrumenty elektroniczne pokazały się też u nas. Pod tym względem nie odstawiamy od reszty świata.

— **Jak się gra?**

— Dobrze. Ale ty zadajesz mi pytania, na które mogę odpowiadać stereotypami. Dlatego pozwól, że przedstawię trochę teorii. Obecnie muzyka jazzowa nie cieszy się tak wielką popularnością jak kiedyś. Dlatego utrzymywanie dużego zespołu staje się trudne.

— **Komputer jest dobry na kryzys?**

— Tak. Mogłem nabyć kilka instrumentów, które w zasadzie zastępują orkiestrę. Współczesna elektronika doprowadziła do tego, że można imitować głos każdego instrumentu. Ba, tworzyć nawet nowe barwy, które do tej pory nie były znane.

— **Naprawdę?**

— Dzięki komputerowi można nawet rozkładać poszczególne dźwięki na różne wysokości — jest to szczególnie pomocne w czasie komponowania nowych utworów. Zapamiętuje wszystko to co ja zagram. Działa równocześnie tak jak magnetofon. Na jednym śladzie nagrywam perkusję, na drugim kontrabas, na trzecim smyczki, na czwartym gitarę, na piątym fortepian, na szóstym trąbki, na siódmym jakieś instrumenty syntezatorowe. Komputer to wszystko razem miesza i po chwili mogą odtworzyć i przesłuchać nowy skomponowany przeze mnie utwór.

— **Z komputerem pod pachą biegasz więc na koncerty?**

— W zasadzie mogę. Ale normalnie przegrywam skomponowany utwór na taśmę magnetofonową, zanoszę do radia i wielu słuchaczy może w ten sposób nagrane moje kompozycje usłyszeć w programie I i III. W podobny sposób komponuję podkład muzyczny do filmów.

— **Czy w czołówce filmowej podajesz również jako współautora nazwę komputera?**

— Nie. Kompozycja jest przecież moja. Została przez komputer tylko wzbogacona poprzez rozłożenie dźwięków takie jakie ja wymyśliłem.

— **Jak więc pracujesz?**

— Całą aparaturę mam w domu. Miałem sporo różnego rodzaju pomysłów muzycznych, których z braku czasu nie mogłem zrealizować. Teraz mam czas, na przykład późnym wieczorem. Siadam, komponuję. Słucham. Jeśli coś mi się nie podoba to mogę poprawić, cofnąć, zmienić. Pozamieniać głosy, pozamieniać takty, powtórzyć, a kiedy jest to zupełnie niedobre kasuję. Dzieje się to tak niby jak na magnetofonie, ale na magnetofon nagrywałem cały utwór i mogłem go najwyżej skasować. Tu jest inaczej. I to jest sukces nowoczesności.

— **Twoją pracę można było oglądać na wystawie, którą zorganizowała firma Atari w czasie festiwalu Jazz Jamboree.**

— Tak. Na komputerze Atari pokazywałem swoją pracę. Wielu muzyków, którzy oglądali pokaz było zachwyconych możliwościami komponowania przy pomocy komputera. Fascynowało ich to, że nuty, które widoczne były na ekranie można było przy pomocy myszki poprawiać i zmieniać, zwiększać lub zmniejszać ich wartości. Jest to naprawdę prawdziwa rewolucja muzyczna i niezwykła pomoc dla kompozytorów.

— **Z pewnością niedługo sam komputer będzie komponował nowe utwory!**

— Nie. Często spotykam się z zarzutem, że komputer tworzy muzykę nieludzką, niehumanistyczną, muzykę dla robotów. Nieprawda. Komputer odtwarza tylko to co ja zagrałem. Odtwarza to w sposób idealny.

— **Tylko?**

— Kiedyś, kiedy nie pracowałem z komputerem i komponowałem muzykę dla swojego zespołu to grałem ją na fortepianie, zapamiętywałem i potem dopiero zapisywałem prostymi nutami na papierze. Natomiast obecnie pracując z komputerem mam, jak już mówiłem, ułatwione zadanie. Komputer jednocześnie zanotuje to wszystko na papierze, to co zagrałem. Często, kiedy patrzę na zapis nutowy swojej kompozycji, jestem zdumiony, że napisałem taki skomplikowany utwór. Nieraz wydaję komputerowi rozkaz by uprościł zapis, żeby nie było tyle łuków, tyle szesnastek czy trzydziestodwojek tylko same ósemki lub ćwiartki. I komputer robi to.

— **Jak widzę zaprzyjaźniłeś się z komputerem.**

— Jest to istotnie rzecz fascynująca. Jeszcze raz powtarzam, że nigdy bez ułatwień jakie daje komputer tak wiele nie skomponowałbym w ostatnim czasie. Program opracowany na Atari pozwala na wiele.

— **Z tego wynika, że ja nie zostanę kompozytorem. A myśląłem, że kupię sobie komputer oraz programy, usiądę i zacznę grać.**

— Prymitywnym kompozytorem możesz zostać. Jeśli pomylisz prostą frazę ze trzy razy i każesz kompute-

rowi wyrównać i poprawić błędy i potem każesz powtórzyć to dziesięć razy z różnego rodzaju instrumentami perkusyjnymi to być może że uda ci się skomponować jakiś utwór, który można zaprezentować w dyskotekce. Tylko, że z mojego punktu widzenia będzie to muzyka nie ludzka i bezwartościowa. Ale idąc za tokiem twego rozumowania na dobrą sprawę dzięki komputerom każdy może zostać kompozytorem.

— **No właśnie...**

— Tak samo każdy może zostać wirtuozem. No, może nie każdy, bo jeśli w ogóle nie gra na żadnym instrumencie, to nie mógłby. Ale jeśli już gra, nawet bardzo powoli jakąś melodię, lub jakąś frazę a potem puścić to szybciej, i naprawdę będzie to wykonanie, którego nie powstydzi się zawodowy muzyk.

— **I nie widzisz w tym żadnego niebezpieczeństwa?**

— Tak. W tym tkwi pewne niebezpieczeństwo. I dużo właśnie takiej tandety zaczyna się szerzyć w dyskotekach na Zachodzie. Ja liczę na to, że nawet człowiek, który na co dzień posługuje się komputerem, zachowuje w swojej pamięci prawdziwą muzykę — Chopina, Pendereckiego...

Myszę, też, że najbardziej interesującym zadaniem tych, którzy programują jest tworzenie oryginalnego brzmienia instrumentów, które znamy od setek lat, skrzypiec czy wiolonczeli. I z tego najlepiej wynika, że człowiek nie zatracił jeszcze poczucia rzeczywistych dźwięków, które płyną z natury. Naturalny dźwięk naturalnego instrumentu z komputera. To jest to o co biją się najlepsi programiści.

Rozmawiał:  
Kazimierz Treger

# KOMPUTERZE

## — WSZYSTKIEGO NAJLEPSZEGO!

**Nie będziemy drugą Japonią. Nie zapowiada się nawet byśmy mieli ambicje i możliwości stać się drugim Tajwanem — przynajmniej, jeśli wziąć pod uwagę rolę, jak w gospodarce obu krajów zamierza powierzyć się informatyce i przemysłowi produkującemu na jej potrzeby. Dla nas oznacza to, że długo jeszcze w „Bajtku” nie pojawią się klany komputerów o nazwach bardziej swojskich niż „Atari” czy „Commodore”.**

Do takich, niezbyt dla fanatyka komputeryzacji optymistycznych, wniosków skłania lektura przyjętego w końcu 1987 roku przez Komitet d/s Nauki i Postępu Technicznego „Programu rozwoju informatyki do roku 1995”. Warto jednak poznać jego założenia. Choćby dlatego, że, jak zapewniają autorzy, jego założenia stanowią maksimum tego, co można osiągnąć w roku 1995.

Co konkretnie przewiduje „Program...”? Czy ci, którzy m.in. dzięki „Bajtkowi” pokonują pierwsze szczeble informatycznego tajemniczenia będą mogli w przyszłej pracy zawodowej wykorzystać nabyte wiadomości?

Program, nieprzypadkowo chyba wybija na plan pierwszy bardzo konkretne zastosowania informatyki jako narzędzia dla zwiększenia efektywności całej gospodarki. Dlatego też przewiduje się przede wszystkim zastosowanie komputerów w systemach wspomagania prac inżynierskich i projektowych, badań naukowych a także procesów produkcyjnych i pomiarowo-kontrolnych. Każde biuro projektowe połowy lat dziewięćdziesiątych dysponować ma systemem komputerowym. Również do wszystkich nowocześniejszych zakładów wprowadzone być mają systemy CAM, zaś licencje produkcyjne nabywane będą także łącznie z odpowiednim hardwarem i softwarem.

Dla realizacji tylko tych trzech zadań potrzeba będzie ok. 250 tys. mikrokomputerów. 550 tys. dalszych otrzymać powinna oświata i szkolnictwo wyższe, a 40 tys. służba zdrowia. W sumie w Polsce połowy lat dziewięćdziesiątych pracować ma prawie pół miliona profesjonalnych mikrokomputerów, drugie tyle edukacyjnych i 400 tysięcy domowych. Będzie zatem z czego korzystać i w szkole, i w pracy. Tyle, że taki poziom rozwoju informatyki oznacza postawienie się na równi z krajami kapitalistycznymi w przypadku... zatrzymania na 8 lat wskazówek zegara.

Najwięcej kłopotów sprawia i sprawiać będzie jeszcze długo technika i technologia. Program zakłada wprawdzie wprowadzanie do produkcji coraz to nowszych typów mikrokomputerów, ale „pewniaków” jest w nim tylko dwóch: „Mazovia-1016” oraz „Elvro-800”. Zważywszy na poziom nowoczesności obu tych konstrukcji wydaje się nieporozumieniem zakładanie, że będzie można je produkować i przede wszystkim eksportować jeszcze w 1995 roku. Na dodatek wszystkie wymienione w programie komputery powinny już być wedle harmonogramu wdrożone do produkcji seryjnej. Tymczasem część jeszcze nie urz-

ła światła dziennego, natomiast przyszłość podstawowego komputera edukacyjnego — „Elvro 800 Junior” jest w momencie, gdy piszę te słowa wciąż mocno niepewna — ruszyła produkcja, lecz jednocześnie utknęła sprzedaż. Od listopada 1987 do połowy stycznia br. Ministerstwo Edukacji Narodowej nie wykupiło już gotowych komputerów. Wygląda zatem, że początki realizacji programu są bolesne. Oby nie okazał się jeszcze jedną papierową wizją.

Oczywiście produkcja jednostek centralnych to jeszcze nie wszystko. Potrzeba monitorów, klawiatur, drukarek, stacji dysków, winchesterów, ploterów itd. Jak na razie w ofercie przemysłu państwowego pustka. Są już natomiast liczne Centralne Programy Badawczo-Rozwojowe w wyniku których powstać mają prototypy nowych urządzeń. Ich wdrożenie nie będzie jednak tak szybkie, jakby się chciało — np. na twarde dyski rodzimej produkcji przyjdzie nam czekać do 1991 roku. Najwyraźniej jednak nie liczymy na żadną długofalową współpracę, czy też kooperację, zarówno ze Wschodem, jak i z Zachodem. Program bowiem przewiduje niemal własny model rodzimego przemysłu komputerowego. Jeśli coś nawet importujemy, to tylko przejściowo. Docelowo produkować mamy wszystko. Pytanie tylko: po co? Czy nie lepiej byłoby spróbować kooperacji, tym bardziej, że np. Węgrów, czy Czechosłowaków też z pewnością nie stać na samodzielne rozwiązanie wszystkich problemów technicznych i konstrukcyjnych w zakresie hardware'u.

Znaków zapytania jest zresztą więcej. Dotyczą one m.in. rozwoju sieci teleinformatycznych — program mówi o tym dość enigmatycznie, czy wreszcie, co bardziej nas chyba jeszcze interesuje, produkcji komputerów domowych. Z zapisów programu wynika, że wciąż jeszcze głównym źródłem zaopatrzenia prywatnych hobbystów w komputery pozostanie „Pewex” i import prywatny. Zakłada się jedynie wprowadzenie na rynek nadwyżki mikrokomputerów edukacyjnych. Czyżby autorzy nie wiedzieli, że jak na razie na świecie tylko nieliczne konstrukcje sprawdzają się w szkole równie dobrze jak w domu?

Podczas posiedzenia KNIPT zatwierdzającego program uzasadnione kontrowersje wzbudziło pominięcie de facto w programie całego ruchu klubów komputerowych — i tego związanego z organizacjami młodzieżowymi, i „nieorganizowanego”. Zwrócił na to uwagę sekretarz generalny Rady Krajowej TMMT Wojciech Wyszomirski. Na jego wniosek opracowany ma być stosowny rozdział programu poświęcony właśnie społecznemu ruchowi komputerowemu.

Wszystkie plany, programy i zamierzenia dotyczące polskiej komputeryzacji mają jednak wielkie „ale”. Możemy produkować mikrokomputery, drukarki, plotery, stacje dysków, ale potrzebne są podzespoły. Dopóki nie podejmie się dramatycznego wręcz wysiłku dla modernizacji CEMI i pozostałych zakładów podzespołowych i nie rozpocznie własnej produkcji choćby części potrzebnych „kości” parafrazując Sienkiewicza powiedzieć można: „Szkoda howoryty, szkoda pisaty”. A na podjęcie działań rzeczywiście nadzwyczajnych jakoś się nie zanoszą. Sumaryczne wielkości nakładów na przemysł komputerowy w latach 1987–1995 będą porównywalne z kosztami budowy jednej kopalni. A to mimo wszystkich deklaracji „Programu...” stawia go raczej w sferze życzeń. A zatem wszystkiego najlepszego polska komputeryzacji!

*Grzegorz Onichimowski*

O ZMIENNYCH DYNAMICZNYCH  
O ZMIENNYCH DYNAMICZNYCH

SŁÓW KILKA

**Zwolennikom i przeciwnikom Pascal-a z pewnością obrzydło już nieustanne wyliczanie zalet tego języka na łamach pism komputerowych (Znacie? No to posłuchajcie: modularność, przejrzysta struktura programów, definiowanie typów danych, zasada predefinicji pojęć, możliwość użycia rekurencji...)**

**Autorzy hymnów na cześć Pascal-a rzadko kiedy dostrzegają jednak pewną inną cechę tego języka, jaką jest możliwość użycia zmiennych dynamicznych. Rzecz to niebanalna, a całkiem obca nawet najwspanialszym wersjom Basic-a czy Fortranu.**

### O CO CHODZI?

To znaczy — co to są owe zmienne dynamiczne? Otóż wiemy, że w Pascal-u mamy możliwość deklarowania dwóch rodzajów zmiennych — globalnych i lokalnych. Zmienne takie istnieją tylko w trakcie wykonania części programu, w której zostały zadeklarowane. Oznacza to, że w momencie, gdy rozpoczyna się wykonywanie np. procedury, program rezerwuje sobie odpowiedni obszar w pamięci dla przechowywania jej zmiennych lokalnych. Wraz z zakończeniem wykonywania procedury obszar ten jest zwalniany i może być wykorzystany do innych celów. Oczywiście tracimy przy tym wartości owych zmiennych.

W odniesieniu do takich zmiennych ukuto nazwę „statyczne”, ponieważ istnieją one w pamięci przez cały czas wykonania tego modułu, w którym je zadeklarowaliśmy. Być może — rozumiemy więc — zmienne dynamiczne to takie zmienne, które możemy tworzyć i usuwać według własnego uznania, w dowolnym momencie trwania programu? Tak, o to właśnie chodzi.

Ale po co?

Oto przykład.

Wyobraźmy sobie zapisany na dysku plik, który zawiera pewne rekordy. Jak wczytać jego zawartość do pamięci? Gdybyśmy dysponowali tylko zmiennymi statycznymi, należałoby umieścić kolejne wczytywane rekordy w tablicy. No tak, ale każdy z łatwością dostrzeże tu dwie kardynalne wady:

1. Rozmiar tablicy określa maksymalną liczbę rekordów, jaka może być wczytana. Plik natomiast może zawierać więcej rekordów;

2. Z drugiej strony, plik może zawierać bardzo mało (nawet zero) rekordów. W takim wypadku duża część naszej tablicy jest bezużyteczna. Nie ma strachu, jeżeli nasz program używa tylko jednej takiej tablicy. Co jednak zrobić, kiedy chcemy jednocześnie operować kilkoma takimi plikami i

# PROGRAMOWAĆ MOŻE KAŻDY

pomieścić je w pamięci jednocześnie? Na zadeklarowanie kilku wystarczających rozmiarami tablic musieliśmy poświęcić ogromne ilości pamięci.

Widzimy tu wyraźnie sens użycia zmiennych o rozmiarach, które mogliśmy regulować w miarę naszych bieżących potrzeb. Zastosowań dla zmiennych dynamicznych można znaleźć wszakże daleko więcej — stwarza je praktyka.

Wiemy już zatem o co chodzi, pozostaje jeszcze pytanie

## JAK?

Kluczem do zrozumienia mechanizmu posługiwania się zmiennymi dynamicznymi jest pojęcie (znów termin ukuty dość dawno) wskaźnika. Wyobraźmy sobie następujące typy danych:

**TYPE** NaszRekord = **RECORD**

Wys, Szer, Dlug: **INTEGER**;

Masa: **REAL**;

**END**;

Wskaźnik = ↑ NaszRekord;

**VAR** P: Wskaźnik;

Definicja typu NaszRekord jest jasna. Wątpliwości budzi definicja typu Wskaźnik (ta podejrzana strzałka...). Czym jest zmienna P, należąca do tego typu?

Strzałka w definicji typu Wskaźnik oznacza, że zmienne tego typu mogą wskazywać na obiekty typu NaszRekord — wymienionego zaraz za strzałką. Co to znaczy „wskazywać”? To znaczy, że wartością takiej zmiennej może być adres (nie wnikamy w to, jak jest on reprezentowany) pewnego, znajdującego się w pamięci rekordu typu NaszRekord. Możemy tę sytuację przedstawić sobie na rysunku (rys. 1).

Tenże wskaźnik stanowi właśnie narzędzie do posługiwania się zmiennymi dynamicznymi. Jak się to robi? To proste: oto podstawowe operacje, jakich możemy dokonywać na wskaźnikach.

1. Możemy je porównywać, ale tylko przy pomocy operacji = i <>. Równość oznacza, że wskaźniki wskazują na ten sam obiekt, nierówność — że na różne.

2. Musimy mieć jakąś możliwość wyrażenia faktu, że wskaźnik nie wskazuje na żaden obiekt znajdujący się w pamięci. W tym celu Pascal zawiera stałą standardową NIL. Oznacza ona po prostu „adres nieistniejący”.



Oto przykłady użycia:

P := NIL;

**IF** Q = NIL **THEN**...

3. Na zmienną będącą wskaźnikiem możemy podstawiać NIL lub wartość dowolnej innej zmiennej — byle należała ona do tego samego typu wskaźnikowego (wskaźnik wskaźnikowi nierówny — zależnie od tego, na jakiego typu obiekt wskazuje).

P := Q; Q := NIL;

4. Ważna operacja:

**NEW** (P);

jest standardową procedurą Pascal-a. Jej działanie jest następujące: zostaje utworzony w pamięci nowy obiekt typu wskazywanego przez parametr, a następnie parametr zaczyna wskazywać na ów obiekt. W powyższym przykładzie w pamięci zostanie zarezerwowane miejsce na nowy rekord typu NaszRekord, zaś zmienna P zaczyna to miejsce — inaczej: nowo utworzony rekord — wskazywać.

5. No tak, ale ja teraz dostać się do poszczególnych pól nowego rekordu? Skoro P nam go wskazuje, pozostaje tylko użyć następujących oznaczeń:

P ↑ — oznacza obiekt wskazywany przez P, a zatem cały rekord typu NaszRekord.

P .Masa — oto, jak dostajemy się do poszczególnych pól wskazywanego rekordu. Gdyby obiektem wskazywanym była np. tablica, to prawidłowy mógłby być np. zapis P ↑ [10]. Oto kilka przykładów:

P ↑ .Masa := 1.05;

**WITH** P ↑ **DO BEGIN**

Wys := 100;

Szer := 10;

**END**;

P ↑ .Dlug := P ↑ .Szer;

6. Musimy jeszcze mieć możliwość usunięcia wybranych struktur z pamięci. Realizację takiego żądania zapewnia standardowa procedura DISPOSE. Jej parametrem jest znów wskaźnik.

**DISPOSE** (P);

Procedura ta powoduje, że zostaje usunięty z pamięci obiekt wskazywany przez P (to znaczy, że zajmowana przezeń pamięć jest zwalniana).

Uzbrojeni w ten zestaw operacji możemy zabrać się do pracy. Na początek zastanówmy się nad problemem przechowywania w pamięci ciągu rekordów, który postawiliśmy na początku.

sposób oszczędny?

Sęk w tym, aby zbudowana struktura danych zajmowała w pamięci tylko tyle miejsca, ile potrzebne jest dla wczytanych rekordów — nie więcej.

Zaproponujemy tu rozwiązanie, które w praktyce okazuje się najbardziej ekonomiczne — mianowicie uszeregowanie naszych rekordów w listę. Czym jest lista? To proste: zauważmy, że wewnątrz rekordu NaszRekord możemy umieścić pola dowolnych typów. Zadeklarujmy tam więc dodatkowo pole typu Wskaźnik. Wtedy rekord typu NaszRekord będzie mógł sam zawierać wskaźnik do innego rekordu, ten zaś do innego, ten do innego... i tak dalej, aż w końcu wskaźnik w którymś rekordzie byłby równy NIL — i na tym lista by się zakończyła. Dla lepszego wyobrażenia sobie takiej sytuacji mamy rysunek 2. Pokazuje on dodatkowo, że zachodzi potrzeba posiadania wskaźnika do pierwszego rekordu takiego ciągu. W przeciwnym wypadku nie mielibyśmy dostępu ani do niego, ani do żadnego z pozostałych rekordów listy.

Spróbujmy zatem zdefiniować odpowiednie typy danych.

**TYPE** Wskaźnik = ↑ NaszRekord;

NaszRekord = **RECORD**

Wys,Szer,Dlug : **INTEGER**;

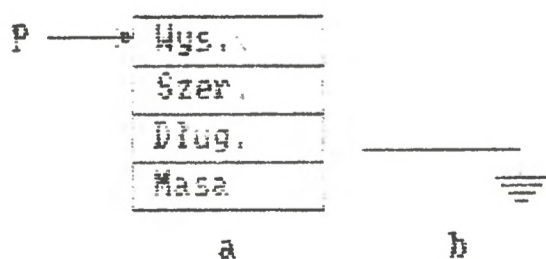
Masa : **REAL**;

Nast : Wskaźnik

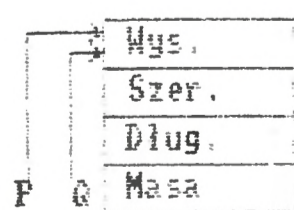
**END**;

**VAR** Początek, P, Q : Wskaźnik;

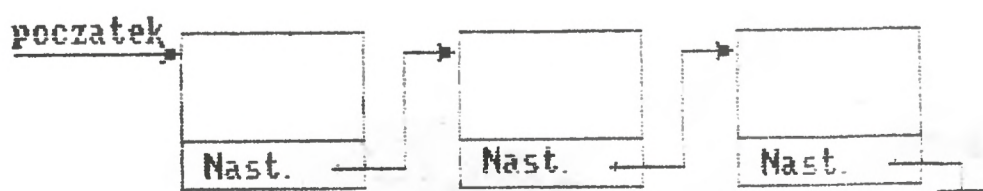
Zaraz, zaraz — powie zaraz spozstrzegawczy czytelnik — w tej definicji typów nie jest spełniona zasada predefinicji, bo typ NaszRekord jest użyty ZANIM w tekście wystąpi jego definicja. Czytelnik będzie miał rację. Co



rys. 1



rys. 2



rys. 3

## STRUKTURY LISTOWE

Jak zatem reprezentować w pamięci taki ciąg rekordów, postępując w

# PROGRAMOWAĆ MOŻE KAŻDY

więcej, w tym konkretnym przypadku nie jest możliwy taki zapis powyższego fragmentu programu, aby zasada definiowania pojęć przed ich użyciem była zachowana w pełni. Dlatego też mamy tu do czynienia z JEDYNYM dopuszczonym w Pascal-u odstępstwem od zasady predefinicji. Mówiąc ściślej, typ wskaźnikowy wolno w tym języku zdefiniować zanim zdefiniuje się typ obiektu wskazywanego.

Skoro już zdefiniowaliśmy typ takiej struktury — jak ją samą utworzyć?

Potrzebna będzie nam jedna zmienna typu Wskaźnik, która pokazywałaby stale na pierwszy z ciągu rekordów. Do wykonywania operacji na rekordach dalszych możemy użyć pomocniczych zmiennych wskaźnikowych, które wskazywałyby na rekordy aktualnie przez nas przetwarzane. Można też postąpić inaczej i w celu uzyskania dostępu do rekordów wewnątrz listy posłużyć się algorytmami rekurencyjnymi. Program przykładowy zamieszczony obok pokazuje, że ich użycie aż się prosi w tym przypadku.

Program ten stanowi rozwiązanie postawionego na początku artykułu problemu na przykładzie prostego notesu telefonicznego. Do wykonywania wszelkich operacji na liście rekordów użyto procedur rekurencyjnych — warto je przeanalizować. Podobne algorytmy, zapisane nierekurencyjnie, znaleźć można w książce N. Wirtha „Wstęp do programowania systematycznego”.

Na zakończenie kilka praktycznych uwag, dotyczących posługiwania się zmiennymi wskaźnikowymi.

1. Trzeba pamiętać, że wskaźnikom nie jest nadawana żadna inicjalna wartość. W szczególności należy pamiętać, aby wskaźnikom nadawać na początku programu wartość NIL. Odwołanie poprzez wskaźnik o nieokreślonej wartości prowadzi może do trudnych do wykrycia błędów.

2. Należy pamiętać o tym, by nie dopuścić do istnienia obiektów dynamicznych nie wskazywanych przez żaden wskaźnik. Do takich obiektów nie sposób się dostać; również nie można ich usunąć procedurą DISPOSE.

3. Jeżeli tworzymy nowy obiekt wskazywany przez NEW (P), to przed następnym wywołaniem NEW (P) musimy zadbać o zachowanie wartości P, które jest wtedy wskaźnikiem do pierwszego utworzonego obiektu. W przeciwnym przypadku ów obiekt znajdzie się w sytuacji z punktu 2.

4. Trzeba uważać, by nie odwoływać się do obiektu wskazywanego przez zmienną, gdy jest ona równa NIL. W wielu przypadkach narzuciłoby się np. użycie takiej instrukcji:

```
IF (P<>NIL) AND (P↑.Masa = 0) THEN...
```

...nie jest to jednak prawidłowe. Pascal wartościuje bowiem wszystkie człony wyrażenia logicznego i z tego powodu będzie starał się obliczyć P↑.Masa niezależnie od tego, czy P jest równe NIL, czy też nie. Prawidłowo należy tę instrukcję rozbić na dwie:

```
IF P<>NIL THEN
```

```
IF P↑.Masa = 0 THEN...
```

5. Należy uświadomić sobie, że in-

strukcja przypisania w odniesieniu do wskaźników ma nieco inne skutki, niż do tego przywykliśmy. Normalnie bowiem efektem przypisania

```
Q := P;
```

jest skopiowanie wartości zmiennej P do zmiennej Q. W przypadku, gdy P i Q są zmiennymi wskaźnikowymi dzieje się podobnie, ale kopiowane są wskaźniki, a nie obiekty wskazywane! Jeżeli np. P wskazuje pewien rekord, to efektem powyższego przypisania będzie sytuacja jak na rys. 3.

W tej sytuacji instrukcja np. P.Masa := 0 zmienia jednocześnie wartość pola Q.Masa — bo przecież w gruncie rzeczy są to te same pola! Jeszcze gorzej jest w przypadku, gdy wykonamy DISPOSE (Q). Wskaźnik P nie zmienia przy tym oczywiście swojej zawartości, ale dalsze odwoływanie się do wskazywanego przezeń rekordu nie ma sensu i może prowadzić do przykrych błędów. Dlatego też w przypadku usuwania obiektu wskazywanego przez wiele wskaźników należy je wszystkie wyszukać i nadać im wartość NIL.

Sądzę, że publikowany program stanowi przykład rzeczywistej wygody, jaką daje posługiwanie się listami. Dla porównania proponuję przejrzeć artykuł „Naprawdę nie święci garnki lepią” z nru 3/87 „Bajtki”. Opisane tam proste bazy danych spełniają funkcje podobne do naszej — ale dla uzyskania podobnego efektu w Basic-u trzeba było uciekać się do niewygodnych metod typu przechowywania rekordów w tablicy — a o wadach tego rozwiązania już wiemy.

Program został napisany w Turbo Pascal-u. Ponieważ opis tego języka był już w „Bajtku” publikowany, sądzę, że w razie potrzeby da się on bez trudu przełożyć na inny dialekt Pascal-a.

Marek Wyrwidąb

## LITERATURA

1. Niklaus Wirth „Wstęp do programowania systematycznego”, wyd. I, WNT 1982 — seria BIO;
2. M. Iglewski, J. Madey, S. Matwin „Pascal”, wyd. III, WNT 1984 — seria BIO.



```
PROGRAM NotesTelefoniczny;
(* Implementacja podstawowych operacji *)
(* na liście, na przykładzie procedur *)
(* obsługi prostej bazy danych. *)
(* Przykład użycia algorytmów rekurencyjnych do obsługi list. *)
(* B A J T E K 1988 *)
(* *)
TYPE
  Slowo = string [20];
  Numer = string [7];
  Wsk = ^Zapis;
  Zapis = record
    Nazwisko:Slowo;
    Telefon:Numer;
    Nast :Wsk;
  end;
VAR
  PoczNotesu:Wsk;
  F :text;
  Lancuch :Slowo;
  Menu;
  Koniec :boolean;
  Zn :char;
  Nazw :Slowo;
  Nowy :Wsk;
  PROCEDURE Wyswietl (P:Wsk);
  begin (* Wypisanie zawartosci listy *)
    if P = nil then
      writeln ('Nie ma (więcej) rekordow.')
    else begin
      write (P^.Nazwisko);
      write (' ':20-Length (P^.Nazwisko));
      writeln (P^.Telefon:7);
      Wyswietl (P^.Nast);
    end;
  end;
  PROCEDURE Szukaj (P:Wsk; Kogo:Slowo);
  begin (* Wyszukiwanie wskazanego rekordu *)
    if P = nil then
      writeln ('Poszukiwanie bez rezultatu.')
    else
      if P^.Nazwisko = Kogo then
        writeln (P^.Telefon:10)
      else
        Szukaj (P^.Nast, Kogo);
    end;
  end;
  PROCEDURE Wstaw (VAR P, Nowy:Wsk);
  begin (* Wstawienie nowego rekordu *)
    (* w porządku alfabetycznym *)
    if P = nil then begin
      Nowy^.Nast := P;
      P := Nowy
    end
    else
      if P^.Nazwisko < Nowy^.Nazwisko then
        Wstaw (P^.Nast, Nowy)
      else begin
        Nowy^.Nast := P;
        P := Nowy
      end
    end;
  end;
  PROCEDURE NowyZapis (VAR Nowy:Wsk);
  begin
    new (Nowy);
    write ('Nazwisko ? ');
    read (Nowy^.Nazwisko); writeln;
    write ('Telefon ? ');
    read (Nowy^.Telefon); writeln;
  end;
  PROCEDURE Usun (VAR P:Wsk; Kogo:Slowo);
  VAR Pom:Wsk;
  begin (* Usunięcie wskazanego rekordu *)
    if P = nil then
      writeln ('Brak wskazanego rekordu.')
    else
      if P^.Nazwisko = Kogo then begin
        Pom := P;
        P := P^.Nast;
        dispose (Pom)
      end
      else
        Usun (P^.Nast, Kogo)
      end;
  end;
  PROCEDURE Kasuj (VAR P:Wsk);
  begin (* Usunięcie całej listy *)
    if P = nil then begin
      Kasuj (P^.Nast);
      dispose (P);
      P := nil;
    end
  end;
  PROCEDURE Pisz (VAR P:Wsk; VAR F:text);
  begin (* Zapisanie listy na dysku *)
    if P = nil then
      Close (F)
    else begin
      writeln (F, P^.Nazwisko);
      writeln (F, P^.Telefon);
      Pisz (P^.Nast, F)
    end
  end;
  PROCEDURE Czyt (VAR P:Wsk; VAR F:text);
  begin (* Odczyt listy z dysku *)
    if eof (F) then begin
      P := nil;
      Close (F)
    end
    else begin
      new (P);
      readln (F, P^.Nazwisko);
      readln (F, P^.Telefon);
      Czyt (P^.Nast, F)
    end
  end;
  FUNCTION Potwierdzone:boolean;
  var Zn:char;
  begin
    write ('Na pewno ? (T/N) ');
    read (Kbd, Zn); writeln (Zn);
    case Zn of
      'T','t' : Potwierdzone := true;
    else Potwierdzone := false
    end
  end;
  Begin (* Program udostępniający funkcje *)
    (* obsługi listy *)
    PoczNotesu := nil;
    Koniec := false;
    Menu := true;
    Assign (F, 'NONAME.TEL');
    while not Koniec do begin
      writeln;
      if Menu then begin
        writeln ('-----M-E-N-U-----');
        writeln ('<Spacja> - wyswietlenie');
        writeln ('S - wyszukiwanie');
        writeln ('W - wstawianie ');
        writeln ('U - usuwanie ');
        writeln ('Z - zapis ');
        writeln ('D - odczyt ');
        writeln ('K - kascwanie ');
        Menu := false;
      end;
      write ('Co Kazesz ? ');
      read (Kbd, Zn); writeln (Zn);
      case Zn of
        ' ' :begin
          write ('NAZWISKO');
          writeln ('TELEFON':19);
          writeln;
          Wyswietl (PoczNotesu)
        end;
        'S','s':begin
          write ('Nazwisko ? ');
          read (Nazw); writeln;
          Szukaj (PoczNotesu, Nazw)
        end;
        'W','w':begin
          NowyZapis (Nowy);
          Wstaw (PoczNotesu, Nowy)
        end;
        'U','u':begin
          write ('Nazwisko ? ');
          read (Nazw); writeln;
          if Potwierdzone then
            Usun (PoczNotesu, Nazw)
          end;
        'K','k':if Potwierdzone then
          Kasuj (PoczNotesu);
        'D','d':if Potwierdzone then
          Koniec := true;
        'O','o':if Potwierdzone then begin
          Kasuj (PoczNotesu);
          reset (F);
          Czyt (PoczNotesu, F)
        end;
        'Z','z':begin
          rewrite (F);
          Pisz (PoczNotesu, F)
        end
      else Menu := true
      end;
    end;
    writeln ('Cesc !!!');
  end;
```

## Wspominaliśmy już w „Bajtku” o możliwości wykorzystania portów joysticków jako wyjść do sterowania urządzeń zewnętrznych. Teraz pora na rozszerzenie tego zagadnienia.

W komputerach Atari gniazda joysticków połączone są z układem o nazwie PIA, którym jest standardowy programowalny układ wejścia/wyjścia 6520. Składa się on z dwóch programowalnych portów 8-bitowych oraz dwóch wejść kontrolnych dla każdego portu. Dzięki temu, że oba porty w układzie 6520 są programowalne istnieje możliwość zaprogramowania każdej linii sygnałowej jako wejścia lub wyjścia.

Na początku omówimy możliwości sterowania urządzeń zewnętrznych przez porty joysticków, przy czym interesować nas będzie jedynie port A układu 6520 (port B wykorzystywany jest do zarządzania bankami pamięci i kontroli złącza szeregowego).

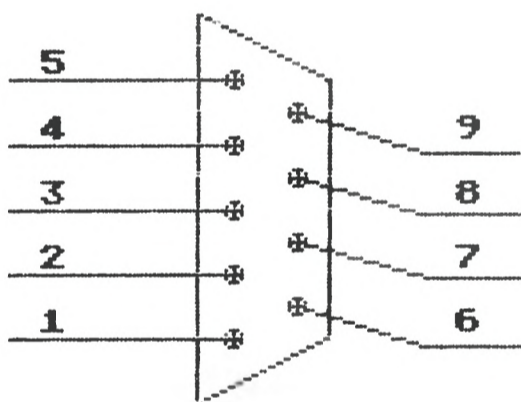
Wstępny opis układu PIA oraz przypisanie poszczególnych bitów portu A układu 6520 do styków przedstawione zostało w „Bajtku” 10/87.

Kontrolę nad portem A uzyskuje się poprzez rejestry PORTA (54016, \$D300) oraz PORTACNTL (54018, \$D302). Wpisując odpowiednie wartości do rejestru PORTACNTL uzyskuje się różne warianty pracy układu 6520.

Znaczenie poszczególnych bitów w tym rejestrze jest następujące:

Bit 0 i 1 — sterowanie wejściem kontroli CA1

### Gniazdo joysticka



Rys. 1. Połączenie PORTA z gniazdami joysticków.

Kolek:	Gniazdo 1	Gniazdo 2
1	bit 0	bit 4
2	bit 1	bit 5
3	bit 2	bit 6
4	bit 3	bit 7
7	+5V	+5V
8	masa	masa

# PORTY JOYSTICKÓW (1)

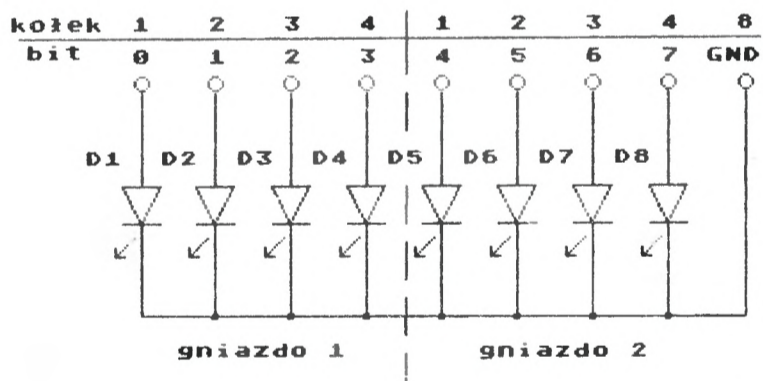
**UWAGA: Nie należy wpisywać liter wydrukowanych przed numerem linii programu. NIE SĄ one częścią linii, lecz kodem kontrolnym „Edytora BASIC-a” (zob. „Bajtek” 1/88 lub „Bajtek-Atari”).**

Bit 2 — sterowanie portem A  
 Bit 3 — 5 — sterowanie wejściem kontroli CA2  
 Bit 6 — zgłoszenie przerwania maskowalnego IRQA1 pochodzące z wejścia CA2  
 Bit 7 — zgłoszenie przerwania maskowalnego IRQA2 pochodzące z wejścia CA1

W tym artykule nie będziemy się zajmować znaczeniem bitów 0, 1, 3-7, interesować nas będzie jedynie bit 2. Jeżeli będzie on równy 0 (mówimy wtedy, że bit ten jest skasowany), to PORTA będzie pracował jako rejestr porządkowania danych. Co to oznacza? Otóż każdemu bitowi tego rejestru odpowiada jedna linia doprowadzona do gniazda joysticka. Jeżeli jakiś bit w tym rejestrze będzie skasowany, to odpowiadająca mu linia w gnieździe joysticka będzie spełniać rolę wejścia. Jeżeli natomiast ustawimy ten bit (zmienimy jego wartość z 0 na 1), to odpowiednia linia w gnieździe joysticka spełniała będzie rolę wyjścia. Gdy bit 2 w PORTACNTL jest ustawiony (1), to dane z gniazda joysticka są przesyłane do PORTA (przy ustawieniu jako wejście) lub z PORTA do gniazda joysticka (przy ustawieniu jako wyjście).

Tyle wstępu, przejdźmy teraz do praktycznego wykorzystania przedstawionych informacji.

Należy przy tym przestrzec przed niestarannym lub niedokładnym wykonywaniem opisanych tu rozwiązań, NIEDOPUSZCZALNE SĄ PRZYPADKOWE ZWARCIA, gdyż może to spowodować uszkodzenie układu 6520. Mówiąc prościej: chwila nieuwagi lub niestaranny montaż i pozbywamy się komputera na kilkanaście dni tracąc przy tym kilka tysięcy złotych na wymianę układu 6520. Nie polecamy



Rys. 2. Przyłączenie diod do gniazd joysticków (D1-D8 — dowolne diody elektroluminescencyjne typu CQYP).

więc samodzielnych prób osobom, które nie mają doświadczenia w montażu układów elektronicznych.

Do podłączenia układów do komputera najlepiej używać typowych kabli z wtyczkami z uszkodzonych joysticków. Można też wykorzystać polskie złącza tzw. komputerowe nr. 871 (ELTRA).

Do gniazda 1 joysticka podłączone są 4 mniej znaczące bity portu A (tzn. bity 0, 1, 2 i 3), pozostałe (4, 5, 6 i 7) do gniazda 2 joysticka. Połączenia te przedstawione są na rys. 1.

Pierwszy układ próbny jest przedstawiony na rys. 2. Spróbujmy go zmontować i podłączyć do portów joysticków.

Niektórzy mogą być zdziwieni brakiem rezystorów szeregowych w układzie, ale gniazda joysticków połączone są wewnątrz komputera z układem 6520 przez rezystory o wartości 2,2 kOhm. Rezystory te zabezpieczają układ 6520 przed przeciążeniem prądowym, ich duża wartość powoduje, że diody będą świecić bardzo słabym światłem. Znajdzie się na to rada, ale o tym nieco później.

Po podłączeniu układu włączamy komputer i wpisujemy poniższy program.

```

GG 10 GRAPHICS 18
CY 20 P=PEEK(54018):POKE 54018,P-4:REM PORT A JAKO REJESTR PORZADKOWANIA DANYCH
IF 30 POKE 54016,255:REM WSZYSTKIE BITY PORTU A USTAWIONE JAKO WYJSCIA
PV 40 POKE 54018,P:REM ODTWORZENIE POPRZE DNIEJ WARTOSCI PORTACNTL
ML 50 FOR I=1 TO 255:POKE 54016,I
QQ 60 POSITION 7,6: ? #6;I;" "
PV 70 FOR K=0 TO 500:NEXT K:REM PETLA DPO ZNIAJACA
IW 80 NEXT I
VK 90 IF PEEK(53279)<>6 THEN 50:REM KONIEC PO NACISNIECIU START
DP 100 P=PEEK(54018):POKE 54018,P-4
YA 110 POKE 54016,0:REM WSZYSTKIE BITY PORTU A USTAWIONE PONOWNIE JAKO WEJSCIA
KD 120 POKE 54018,P
NW 130 END
    
```

Po uruchomieniu programu zaobserwujemy zapalanie się diod w różnej kolejności. Jeżeli diody będą ustawione w takiej kolejności, jak na rys. 2, to obserwować będziemy binarne (dwójkowe) przedstawienie liczby wyświetlanej na ekranie, przy czym dioda zapalona odpowiada logicznej jedynce, a dioda zgaszona logicznemu zeru.

W następnych częściach artykułu przedstawione zostaną rozszerzenia powyższego układu oraz układy bardziej skomplikowane i poważniejsze o praktycznym zastosowaniu.

*Przemysław Strzelecki  
Wojciech Zientara*

# TAJEMNICE ATARI (6)

## Jak na razie dotarł tylko jeden list od czytelników Bajtka, w którym opisana byłaby sytuacja pomagająca wygrać w grze jej uczestnikowi.

Jego autorem jest Maciej Gradowski z Krakowa. Czyżby użytkownicy komputerów ATARI nie byli w stanie odkryć ich sekretów? W to nie wierzymy i w dalszym ciągu czekamy na listy z „tajemnicami”.

Tymczasem podajemy kody niezbędne do teleportacji w najnowszym przeboju komputerowej giełdy — grze „Star Quake”. Są to: Delta, Triad, Penta, Kernx, Atari, Whole, Salco, Artic, Minim, Argon, Cosec, Crash, Secon, Z.A.P. i Quark.

W grze „Hacker” na pytanie komputera o kod wywoławczy (LOGON PLEASE) odpowiadamy wpisując za pomocą klawiatury słowo **Australia**. Wykonanie takiego posunięcia uwalnia gracza od przechodzenia przez test robota SRU i pozwala od razu przejść do części zasadniczej gry. Na tym etapie uczestnik zabawy ma cztery szanse, aby zostać złapanym przez obcego satelitę bez ponoszenia konsekwencji tego faktu. Warunkiem determinującym dal-

szą kontynuację gry jest znajomość kodów rozpoznawczych. Oto one:

- pierwszy raz MAGMA, LTD
- drugi raz AX-0310479
- trzeci raz HYDRAULIC
- czwarty raz AUSTRALIA
- piąty raz KONIEC GRY

Dla tych, którzy nie chcą ułatwiać sobie życia i omijać tekstu robota podajemy elementy niezbędne do identyfikacji. Są to:

- INFRARED VIDEO IMAGE SENSOR (ośrodek centralno-wizyjny robota)
- ASYNCHRONOUS DATA COMPACTOR (główny receptor danych dotykowych)
- HYDRAULIC MOTIVATOR (główny ośrodek napędowy robota)
- PHLAMSON JOINT (płazmowy amortyzator kierunkowy)
- THELMAN PORT (złącze urządzeń zewnętrznych).

„Montezuma’s Revenge” jest dla każdego posiadacza Atari czymś w rodzaju standardu. Z reguły od tej gry zaczyna się wspólną zabawę z komputerem. Być może zaszkodzi niektórym, ale jest sposób na to, aby utrudnić przejście Pedra przez tak dobrze znane już komnaty grobowca faraonów. Wystarczy tylko po załadowaniu progra-

mu do pamięci nacisnąć klawisz RESET, a następnie klawisz SELECT. Dzięki temu potroi się liczba śmiercionośnych czaszek na aktualnej planszy gry i dotarcie do skarbów Montezumy nie będzie już dziecinną igraszką dla najbardziej nawet zaawansowanych zawodników.

Ciekawe, czy wiecie jak zwolnić tempo walki w „Karate International”. Rozwiązanie tej zagadki tkwi w klawiszu X. Wystarczy go nacisnąć a następnie wybrać numer od jednego do czterech, pamiętając o zasadzie, że im wyższa cyfra tym wolniejsze ruchy wykonuje kierowana przez joystick postać na ekranie. Dodatkowo w tej wersji gry istnieje możliwość zmiany aktualnej planszy gry. Wykonuje się to za pomocą klawisza OPTION.

W grze pod tytułem „Quasimodo” o ile chce się przejść do dalszych etapów należy w pierwszej komnacie strącić co najmniej trzech łuczniczków wspinających się po drabinach, ale w taki sposób aby żaden następny nie był widoczny. Aby tego dokonać trzeba użyć kul ułożonych w stosy na szczycie muru.

Na ten raz to już wszystko. Dziękujemy „pierwszemu odważnemu” Maćkowi, że do nas napisał i czekamy na jeszcze.

*Sergiusz Piotrowski  
Tomasz Mazur*

## MIGAJĄCY KURSOR

Po zamieszczeniu w ostatnim odcinku cyklu „Nie bój się przerwań” prośby o nadsyłanie własnych proceder otrzymaliśmy od Czytelników kilkanaście listów.

Dziwnym trafem wielu użytkowników Atari przeszkadza brak migotania kursora. Wybraliśmy trzy programy, które usuwają tę niedogodność (?). Dla wszystkich, którzy chcieliby bliżej poznać przerwania Atari, zamieszczamy oprócz wydruków w Basic-u także wydruki programów źródłowych w assemblerze MAC/65. Zostały one odtworzone w redakcji, gdyż autorzy „zapomnieli” je przysłać.

Dwa pierwsze programy wykorzystują przerwanie zegara TIMER 2, jednak w różny sposób. W pierwszym migotanie znaku jest realizowane przez ingerencję w rejestr kontroli wyglądu znaków ekranowych (CHARCNT), drugi natomiast ingeruje w pamięć obrazu przy użyciu rejestru zawierającego adres kursora. Ostatni program wykorzystuje przerwanie synchronizacji pionowej i rejestr zegara systemu. Dodatkowo podczas procedury wywoływanej przez RESET ustawiane są różne parametry komputera. Dzięki temu program ten może być szczególnie przydatny dla osób dużo pracujących z komputerem.  
(redakcja)

Jedną z wad edytora BASIC-a w komputerze Atari 800XL jest brak migającego kursora, co utrudnia jego lokalizację. Aby to poprawić ułożyłem krótki program.

Po jego uruchomieniu kursor zaczyna migać. Prędkość migania można zmienić poprzez zmianę zawartości komórki 1549 (zalecana 20). Kursor przestaje migać po naciśnięciu RESET; aby przywrócić stan poprzedni należy wykonać sekwencję rozkazów z linii 90—120. Kursor nie przestaje migać po zmianie trybu graficznego. Procedura jest relokowalna; adres początku umieszczony jest w komórkach 552 i 553.

Uwaga: przy dużej prędkości wyprowadzania danych na ekran (np. listowanie) kursor może zostawiać ślad.

Bartosz Polednia

```
10 REM *****
20 REM *   MIGAJACY KURSOR   *
30 REM * (C) 1987 B.POLEDNIA *
40 REM *****
50 RESTORE
60 FOR A=1536 TO 1553
70 READ X:POKE A,X
80 NEXT A
90 POKE 552,0:POKE 553,6
100 POKE 538,10:POKE 539,0
110 POKE 54286,64
120 DATA 162,0,161,94,73,128
130 DATA 129,94,73,0,129,94
140 DATA 169,20,141,26,2,96
```

```
0100 ;Migajacy kursor - MAC/65
0110 ;
0120 OLDADR = $5E
0130 TIMCOUNT2 = $021A
0140 *=$0600
0150 ;
0160 LDX #$00
0170 LDA (OLDADR,X)
0180 EOR #$60
0190 STA (OLDADR,X)
0200 EOR #$00
0210 STA (OLDADR,X)
0220 LDA #$14
0230 STA TIMCOUNT2
0240 RTS
```

Program napisany w języku maszynowym wykorzystuje przerwanie VBLK. Na uwagę zasługuje fakt, że program działa dalej po naciśnięciu RESET. Jeśli chcemy wyłączyć pulsowanie kursora, to wystarczy POKE 9,0 i RESET. Powrót do migania uzyskamy przez POKE 9,1 i RESET. Program umieszczony jest na szóstej stronie pamięci.

Adam Hartwig

```
0 REM MIG-CURSOR
10 REM (c)1987 Adam Hartwig
20 FOR X=1536 TO 1578:READ A:POKE X,A:
NEXT X
30 DATA 24,173,243,2,105,2,201,4,208,2
,169,0,141,243,2,169,8,141,26,2,96
40 DATA 169,0,141,27,2,141,40,2,169,6,
141,26,2,141,41,2,169,64,141,14,212,96
50 POKE 539,0:POKE 538,2:POKE 552,0:PO
KE 553,6:POKE 54286,64:POKE 12,21:POKE
13,6:POKE 9,1
```

```
0100 ;Migajacy kursor - MAC/65
0110 ;
0120 TIMVEC2 = $0228
0130 TIMCNT2 = $021A
0140 CHARCNT = $02F3
0150 NMIEN = $D40E
0160 *=$0600
0170 ;
```

```
0180 START CLC
0190 LDA CHARCNT
0200 ADC #$02
0210 CMP #$04
0220 BNE JUMP
0230 LDA #$00
0240 JUMP STA CHARCNT
0250 LDA #$08
0260 STA TIMCNT2
0270 RTS
0280 LDA # <START
0290 STA TIMCNT2+1
0300 STA TIMVEC2
0310 LDA # >START
0320 STA TIMCNT2
0330 STA TIMVEC2+1
0340 LDA #$40
0350 STA NMIEN
0360 RTS
```

Przedstawiony program może być pomocny przy pracy z komputerem, powoduje bowiem migotanie kursora, który staje się przez to bardziej widoczny i łatwiejsze jest redagowanie pisanego tekstu. Oprócz tego po każdym RESET ustawiane są zgodnie z życzeniem użytkownika tak ważne przy długiej pracy z komputerem parametry jak: kolor tła i ramki, jasność tekstu, lewy i prawy margines, a także szybkość wprowadzania znaków z klawiatury. Teraz naciśnięcie (nawet przypadkowe) klawisza RESET nie powoduje irytującej zmiany ustawionych parametrów.

Po uruchomieniu programu mamy migający kursor, tło ciemnozielone, ramkę jasnoniebieską, lewy margines = 2, prawy = 37, czas do pierwszego powtórzenia klawisza = 15, szybkość powtarzania = 2. Jeżeli ustawione wartości komuś nie odpowiadają, można je zmienić dwoma sposobami: w samym programie w liniach DATA (trzeba również obliczyć nową sumę kontrolną i wstawić ją do linii 16 zamiast 8208), albo w trybie bezpośrednim za pomocą instrukcji POKE. A oto lista możliwych zmian:

POKE 1542,41	— kursor migający	linia 20, liczba 41
POKE 1542,9	— kursor normalny	
POKE 1543,x	— częstotliwość migania	linia 20, liczba 8
POKE 1572,x	— lewy margines	linia 24, liczby 2 i 37
POKE 1576,x	— prawy margines	
POKE 1580,x	— jasność tekstu	
POKE 1585,x	— kolor tła	linia 26, liczby 12, 192 i 148
POKE 1590,x	— kolor ramki	
POKE 1595,x	— czas do powtórzenia	linia 28, liczby 15
POKE 1600,x	— szybkość powtarzania	i 2 (po 169)

Marek Krupa

```
10 REM * ACC Szczecin '87 *
12 FOR A=1536 TO 1615
14 READ B:C=C+B:POKE A,B:NEXT A
16 IF C<>8208 THEN ? "Pomyłka w liniac
h DATA !":END
18 A=USR(1554):NEW
20 DATA 169,2,133,9,165,20,41,8,240,2,
169,2,141,243,2,76,98,228
22 DATA 104,169,96,133,0,169,255,133,8
,169,35,133,2,169,6,133,3
24 DATA 169,2,133,82,169,37,133,83
26 DATA 169,12,141,197,2,169,192,141,1
98,2,169,148,141,200,2
28 DATA 169,15,141,217,2,169,2,141,218
30 DATA 2,160,0,162,6,169,7,32,92,228
32 DATA 108,12,0
```





# Polskie

Temat polskich znaków był już w **KLANIE COMMODORE** poruszany niejednokrotnie. Podawany był sposób ich definiowania dla Commodore 64, podawany był także dla C-128 w trybie 40 znaków. Poniżej przedstawiamy Czytelnikom dwa artykuły kończące niejako ten cykl — polskie znaki dla C-16/116+4 oraz dla C-128 w trybie 80 znaków.

# Znaki

## C128

**Mikrokomputery, którymi się posługujemy nie są niestety standardowo zaopatrzone w polski zestaw znaków. Stanowi to często poważne utrudnienie, gdy chcemy by nasz skądinąd świetny program komunikował się z nami w naszym rodzimym języku.**

Na łamach BAJTKA pojawiały się już programy umożliwiające przeddefiniowanie znaków np. w trybie 40 kolumn w komputerze Commodore 128. Wszystkie te metody mają jednak jedną kolosalną wadę: zabierają dużo cennego miejsca w pamięci bo jak wiemy generator znaków, żeby móc go przeddefiniować, należy najpierw przepisać z ROM na RAM. Komputer C-128 może pracować w dwóch trybach: 40 i 80 kolumn. W trybie 40 kolumnowym ekran jest kontrolowany przez układ VIC 8564. Jest to układ prawie identyczny (oprócz dwóch dodatkowych rejestrów) ze znanym z C-64 układem VIC 6564. Używając tego trybu pracy możemy korzystać zarówno ze zwykłego monitora jak i telewizora. Ponieważ komputer C-128 jest już komputerem półprofesjonalnym, ma on możliwość korzystania z drugiego trybu pracy (80 kolumn). Daje to użytkownikom możliwość wygodnego korzystania z całej gamy programów użytkowych jak edytory tekstów, bazy danych oraz z programów pracujących pod kontrolą systemu operacyjnego CP/M. Do tego stało się jednak konieczne zainstalowanie nowego układu VDC 8563 (Video Display Controller). W tym wypadku możliwe jest jednoczesne podłączenie do komputera monitora RGB oprócz jednego z już wyżej wymienionych odbiorników.

Przedstawiony poniżej program oprócz tego, że umożliwia korzystanie z polskich znaków w trybie 80 kolumn, NIE ZUŻYWA ANI JEDNEGO DODATKOWEGO BAJTU PAMIĘCI. Znaczący to tyle, że po wprowadzeniu i uruchomieniu tego programu będziemy dysponowali polskim zestawem znaków oraz taką samą pamięcią jak po normalnym włączeniu komputera. Dzieje się tak, ponieważ w odróżnieniu od VIC 8564, VDC 8563 posiada tylko dwie lokacje w pamięci poprzez które możemy się odwoływać do 37 rejestrów tego układu (adresy \$D600 i \$D601). Oznacza to, że w całej pamięci komputera są tylko dwie komórki poprzez które system operacyjny oraz my mamy wpływ na działalność tego układu oraz na dodatkowe 16K własnej RAM, zupełnie niezależnej od komputera. Właśnie w tym obszarze przechowywany jest zestaw znaków (zupełnie odmienny od tego jakim posługuje się komputer na ekranie 40 kolumnowym). Cecha ta pozwala przeddefiniować znaki, a zarazem zaoszczędzić trochę miejsca. No tak, ale jak to zrobić mając bezpośredni dostęp tylko do dwóch rejestrów bez możliwości posłużenia się instrukcjami POKE i PEEK? Pierwsza część przedstawionego programu (do linii 190 stanowi właśnie rodzaj rozbudowanej instrukcji POKE, która umożliwia odwoływanie się do VDC 8563. Druga część programu zajmuje się natomiast definicją polskich liter. Po uruchomieniu programu polskie znaki uzyskujemy następująco:

```

CMD      -a =ą
CMD      -z =ż
CMD      -c =ć
CMD      -n =ń
CMD      -l =ł
CMD      -e =ę
CMD      -s =ś
CMD      -o =ó
CMD      +  =ż
SHIFT    +  =
SHIFT    + £ =ł
    
```

CMD oznacza klawisz COMMODORE w lewym dolnym rogu klawiatury.

Do zdefiniowania znaków w trybie 80-kolumnowym VDC 8563 rezerwuje 2x64 bitów na każdy znak, chociaż istniejący zestaw jest zdefiniowany na matrycach 8x8 bitów. Taki tryb rozmieszczenia kolejnych znaków, oraz fakt, że VDC 8563 może jednocześnie korzystać z 512 znaków graficznych i literowych powoduje, że należy korzystać z następujących wzorów dla określenia pozycji znaku, który chcemy przeddefiniować:

ASC \* 16 + 8192 dla trybu „dużych liter” (uppercase).  
 ASC \* 16 + 12288 dla trybu „małych liter” (lowercase).  
 W liniach 340-440 pierwsza wartość to kod ASCII przeddefiniowywanego znaku. Pozostałe wartości w każdej linii to dane określające nowy znak.

*Dominik Falkowski*

```

B7 100 REM ** POLSKIE LITERY C-128/80 KOLUMN **
38 105 :
D5 110 REM **          DOMINIK FALKOWSKI          **
    
```

```

36 115 REM **          87/12/28          **
78 120 :
89 125 :
55 130 FAST:PRINTCHR$(14)
D0 135 FOR X=6144 TO 6182
BE 140 READ A:DF=DF+A:POKE X,A
EB 145 NEXT
42 150 IF DF<>3411 THENPRINTCHR$(7);:LIST:END
B3 155 DATA 072,138,072,152,072,169,002,141,040
B2 160 DATA 010,162,018,104,032,027,024,232,104
D1 165 DATA 032,027,024,162,031,104,076,027,024
7B 170 DATA 142,000,214,044,000,214,016,251,141
6F 175 DATA 001,214,096
5F 180 RESTORE 250
09 185 DO
FA 190 READ FG
0A 195 IF FG=-1 THEN EXIT
E5 200 RT=FG*16+12288
BD 205 FOR X=0 TO 7
2B 210 AD=RT+X
55 215 HI=INT(AD/256):LO=AD-(HI*256)
9A 220 READ CH
DF 225 SYS 6144,CH,LO,HI
0B 230 NEXT X
FE 235 LOOP
2B 240 PRINT " ▾ _ _ _ _ _ _ _ _ _ _ | "
2E 245 NEW:END
45 250 DATA 124,004,008,060,102,096,102,060,000
07 255 DATA 113,000,000,060,102,126,096,062,003
95 260 DATA 118,056,024,028,024,056,024,060,000
8C 265 DATA 106,004,008,124,102,102,102,102,000
E3 270 DATA 121,004,008,060,102,102,102,060,000
40 275 DATA 110,004,008,060,096,060,006,124,000
0B 280 DATA 109,004,008,126,012,024,048,126,000
5D 285 DATA 100,000,024,126,012,024,048,126,000
4D 290 DATA 105,096,096,112,096,224,096,126,000
2A 295 DATA 122,024,126,012,024,048,096,126,000
52 300 DATA 112,000,000,060,006,062,102,062,003
39 305 DATA -1
    
```

## C-16/116/+4

**Amator programowania stosunkowo prędko zaczyna zastanawiać się, w jaki sposób wzbogacić czy uczynić program bardziej zrozumiałym poprzez zastosowanie w nim polskich znaków diakrytycznych.**

Generalnie rzecz biorąc we wszystkich komputerach Commodore dokonuje się tego tak samo: oryginalne znaki są najpierw kopiowane z pamięci ROM generatora znaków do pamięci RAM, następnie użytkownik musi je odpowiednio zmodyfikować zgodnie ze swoimi potrzebami. Po tej operacji informuje się procesor graficzny skąd ma czerpać informacje o znakach i... gotowe.

Dla rodziny Commodore 16/116/+4 wszystkie funkcje spełnia poniżej przedstawiony program. Nowe znaki są w nim przypisane następującym klawiszom:

```

ą — CMD + a      A — CMD + Q
ć — CMD + c      Ć — CMD + Y
ę — CMD + e      E — CMD + R
ł — CMD + l      Ł — CMD + K
ń — CMD + n      Ń — CMD + B
ó — CMD + o      Ó — CMD + I
ś — CMD + s      Ś — CMD + W
ż — CMD + z      Ż — CMD + D
z — CMD + x      Z — CMD + F
    
```

Znak „CMD” oznacza, że należy najpierw wcisnąć klawisz Commodore Logo Key (lewy dolny róg klawiatury) a następnie podaną literę.

Pierwsze i ostatnie pytanie zadawane nam przez program odnosi się do miejsca w pamięci w którym zostanie ulokowany nasz zestaw znaków. Posiadacze C-16/116 z

pamięcią nierozszerzoną mogą wpisać tu wartości od 8 do 14; posiadacze PLUS/4 i C-16/116 z pamięcią rozszerzoną — wartości od 8 do 62. Im większa będzie podana przez nas ta wartość tym więcej będziemy mieli pamięci wolnej (nowy zestaw znaków zajmuje 1 KB pamięci). Klawisze funkcyjne f2 i f3 będą pełniły rolę przełącznika — wciśnięcie f2 powoduje wyłączenie zestawu polskich znaków podczas gdy f3 jego włączenie. Jeżeli podczas używania nowego zestawu znaków popełnimy błąd wpisując instrukcję czy rozkaz języka BASIC, to system operacyjny przełączy procesor graficzny tylko częściowo i ekran zablokuje się. Odblokować ekran można wciskając klawisz funkcyjny f2 lub f3.

*Kazimierz Stróż*

```

60 100 SCNCLR
E2 105 T$=CHR$(18)+"POL.LIT"+CHR$(146)
94 110 CHAR 1,16,3,T$
F5 115 CHAR 1,4,7,"PROGRAM GENERUJACY POLSKIE Z
NAKI"
C4 120 CHAR 1,5,9,"DLA COMMODORE 16, 116 I PLUS
/4"
2B 125 PRINT:PRINT
1B 135 INPUT "LOKALIZACJA : ";HX
9B 140 HX=4*HX
5A 145 POKE 55,0:POKE 56,HX
5B 150 HX=PEEK(56):ZN=54272
5A 155 A$="POKE65299,"+STR$(HX)
F5 160 A$=A$+" :POKE65298,192:POKE1351,128"+CHR$(
13)
EF 165 KEY 3,A$
9B 170 A$="POKE65299,208:POKE65298,196:POKE1351
,0"
EB 175 KEY 2,A$+CHR$(13)
4B 180 POKE 1177,62
C5 185 FOR I=0 TO 1023
CB 190 POKE 256*HX+I,PEEK(ZN+I)
AF 195 NEXT
9C 200 POKE 1177,63:POKE 1351,128
50 205 POKE 65298,192:POKE 65299,HX
47 210 FOR J=1 TO 18 : READ B
E3 215 FOR I=0 TO 7 :READ A
6D 220 POKE 256*HX+8*B+I,A
CE 225 NEXT:NEXT
4F 230 :
5A 235 SCNCLR
EF 240 CHAR 1,6,10,"POLSKIE ZNAKI ZAINSTALOWANE
"
14 245 PRINT:PRINT:PRINT
E5 250 END
12 255 :
B3 260 DATA 112,000,000,060,006,062,102,062,004
FD 265 DATA 113,000,000,060,102,126,096,060,008
E3 270 DATA 121,004,008,060,102,102,102,060,000
BB 275 DATA 124,004,008,060,102,096,102,060,000
4C 280 DATA 106,004,008,124,102,102,102,102,000
C9 285 DATA 109,004,008,126,012,024,048,126,000
31 290 DATA 125,008,000,126,012,024,048,126,000
46 295 DATA 118,056,024,028,024,056,024,028,000
90 300 DATA 110,004,008,060,096,060,006,124,000
1A 305 DATA 107,024,060,102,126,102,102,102,008
BF 310 DATA 114,126,096,096,120,096,096,126,004
BB 315 DATA 098,060,106,098,098,098,098,060,000
4F 320 DATA 126,060,106,096,096,096,102,060,000
69 325 DATA 127,106,110,118,126,110,102,098,000
73 330 DATA 108,126,006,012,126,048,096,126,000
63 335 DATA 123,126,038,012,024,048,096,126,000
FB 340 DATA 097,096,104,112,096,224,096,126,000
FD 345 DATA 115,060,106,096,060,006,102,060,000
    
```



## PAPIEROWY IBM

**Nie ma sensu chyba wspominać czym jest firma IBM na rynku komputerowym, gdyż nie ma chyba entuzjasty komputerów, który by o niej nie słyszał. Komputer jednak to nie tylko sprzęt, konieczna jest także dla niego szczegółowa dokumentacja bez której nie możliwe byłoby wykorzystanie wszystkich jego możliwości. W tej zaś dziedzinie odpowiednikiem IBM na rynku wydawniczym jest zachodniemiecka firma DATA BECKER.**

DATA BECKER powinna być dobrze znana polskim czytelnikom, a zwłaszcza entuzjastom mikroinformatyki. Swoją działalność firma rozwija w trzech głównych kierunkach: literatury komputerowej o charakterze ogólnym, literatury o charakterze inżynierskim i elektronicznym oraz produkcji oprogramowania. Z chwilą rozpoczęcia się boomu komputerowego rynek stał się bardzo chłonny na wszelkie publikacje dotyczące mikrokomputerów. Sytuację tę wykorzystał znakomicie szef firmy, dr Achim Becker tworząc podwaliny pod potężne przedsiębiorstwo wydawnicze, które ma już na swym koncie znaczący dorobek. Publikacje takie jak zasady programowania w danym języku, czy programowanie w asemblerze zniknęły szybko z półek księgarskich, z biegiem czasu jednak czytelnik zaczął poszukiwać publikacji i książek bardziej zaawansowanych omawiających mapy pamięci poszczególnych modeli mikrokomputerów, architektury systemów operacyjnych, zasadę działania z punktu widzenia współpracy poszczególnych układów scalonych itp. Elastyczne po-

dejsie do wymagań rynku spowodowało, że DATA BECKER zanotowała następną sukces. Powstało wtedy wiele książek o tej tematyce — np. ATARI ST INTERN, Das Grosse Floppybuch zur 1541 itp. Poszczególne modele komputerów i urządzeń peryferyjnych były „rozpruwane” przez specjalistów — elektroników pracujących dla firmy, a doświadczenia sprzedawane właśnie w formie książek z których sporo to prace zbiorowe.

Trzecim kierunkiem działalności jest oprogramowanie. Myślę, że większość posiadaczy Commodore czy Amstrada zna doskonale programy z końcówką „MAT”, np. TEXTOMAT, DATAMAT, FAKTUMAT, DISKOMAT czy KONTO-MAT. Większość z nich jest dostępna pod takimi samymi nazwami na wszystkich modelach komputerów od Commodore począwszy a na ATARI ST kończywszy. I znów trzeba tu wspomnieć o rozsądnym podejściu szefów firmy: zamiast na ilość DATA BECKER poszła na jakość. Przykładem może być tu np. TEXTOMAT 128, który (nie bez kozery!) jest uważany w Polsce za jeden z najlepszych edytorów tekstu.

Pieniądz robi pieniądz o czym zdają się doskonale wiedzieć szefowie DATA BECKER. Parę lat temu firma ta weszła na rynek amerykański podpisując umowę z firmą ABACUS SOFTWARE, która jest wyłącznym przedstawicielem DATA BECKER w USA. Współpraca ta przynosi wymierne korzyści nie tylko dla obu firm, ale także i dla czytelników w postaci coraz to nowych publikacji.

Poniżej chciałbym przedstawić czytelnikom jedną z książek DATA BECKER, którą redakcja otrzymała do recenzji. Książka ta, autorstwa Froitzheima i Kausmanna zatytułowana jest EFFEKTIV UND KREATIV i poświęcona jest komputerowi Commodore PLUS/4. Pierwsza część książki poświęcona jest oprogramowaniu zawartemu w pamięci ROM tego modelu i szczegółowo wyjaśniania w jaki sposób tworzy się bazy danych, drukuje faktury, miesza grafikę z tekstem czy przenosi dane z bazy do edytora tekstu. Omówienie to zawarte jest na 125 stronach z 244 składających się na całą publikację. W dalszych czterech rozdziałach czytelnik znajdzie 18 programów użytkowych napisanych w całości w BASIC co z kolei pozwala na ich dokładne przeanalizowanie. Programy te są poświęcone sortowaniu danych, obsłudze stacji dysku, tworzeniu „okienek”, wykresów słupkowych i wykresów funkcji itp. Dla każdego coś milego...

Do najciekawszych należy zaliczyć rozdział ostatni w którym przedstawiona została mapa pamięci Commodore PLUS/4 od adresu \$0000 do \$07FC (2044 dziesiętnie), oraz uwagi dotyczące wyboru i uzyskania kolorów. Taki dodatek (a zwłaszcza mapa pamięci) czyni książkę tę o tyle bardziej wartościową, że nawet jeśli komus jest już BASIC dobrze znany, to będzie ona także pomocna przy programowaniu w języku wewnętrznym. Reasumując jest to publikacja starannie przemyślana pozwalająca czytelnikowi znacznie lepiej wykorzystywać możliwości Commodore PLUS/4. Jeżeli dodatkowo weźmiemy pod uwagę, że panuje generalna posucha na literaturę i programy do tego modelu Commodore, książka ta okaże się na pewno bardzo przydatna.

FROITZHEIM KAUSMANN  
EFFEKTIV UND KREATIV  
Wydawca: DATA BECKER, 1985  
244 strony, cena 49 DM,  
ISBN 3-89011-073-8

*Klaudiusz Dybowski*

## PRZEDSTAWIAMY WARSAW BASIC

# JAK ROZBUDOWAĆ INTERPRETER?

**Użytkownik mikrokomputera klasy VIC-20 lub COMMODORE 64/128 bez specjalistycznego przygotowania informatycznego, jest prawdopodobnie przekonany, że napisanie interpretera to zadanie niewykonalne dla hobbystów.**

Na szczęście, w naszym przypadku zabawa w pisanie interpretera nie wymaga tworzenia go od początku, gdyż wystarcza jego modyfikacja. Można ją wykonać metodą małych kroków. Daje to sporo satysfakcji, szczególnie w przypadku wspólnych mikrokomputerów, bowiem CMB V2.0 jest ubogi, ale napisany bardzo elastycznie.

W rozpoczynanym dziś cyklu będziemy mówić i konkretnym rozwiązaniu dotyczącym konkretnego mikrokomputera. Chcemy przedstawić jak jest zrobiony i jak działa Warsaw BASIC. Czytelników, którzy będą próbowali porównywać możliwości tego interpretera BASICa z innym, prosimy o branie przy tym pod uwagę tylko tych ich wersji, które są zaimplementowane na podobnym sprzęcie.

Warsaw BASIC jest interpreterem w pełni profesjonalnym, instalowanym w mikrokomputerze COMMODORE 64, obejmującym wreszcie wszystkie czynności związane z kodowaniem, interpretacją kodu, obsługą przerwań, wyprowadzaniem danych i programu itp. W celu uproszczenia dalszych rozważań wydzielimy spośród wszystkich tych czynności, wykonywanych przez interpreter, dwie podstawowe: kodowanie i interpretację.

### Kodowanie

Etap ten zaczyna się od podania komputerowi treści do przetworzenia i kończy się zapisaniem jej w odpowiednim miejscu w pamięci wewnętrznej komputera.

Jeżeli napiszemy na ekranie np.

(1) 100 a\$ = „Warsaw BASIC”: print a\$  
i naciśniemy klawisz RETURN, to w odpowiednim miejscu w pamięci wewnętrznej (do tego jeszcze wrócimy!) w kolejnych „komórkach” pojawi się następujący ciąg kodów (zapisanych tutaj w systemie szesnastkowym):

(2) 64 00 41 24 b2 22 d8 41 52 53 41 58  
20 c2 c1 c9 c3 22 3a 99 41 24 00

Numer linii 100 reprezentowany jest przez zawartość (w systemie szesnastkowym) nb=64 i wb=00 dwóch pierwszych „komórek”, obliczoną według wzoru  $nb + wb \times 256 = 100$ .

Jak można zauważyć, następnym kolejnym znakom w linii (1) odpowiada jeden kod w linii (2) za wyjątkiem ciągu znaków p r i n t. Zaznaczyła się tutaj działalność odpowiedniej procedury, która rozpoznała ten ciąg znaków i zakodowała go w postaci liczby 99 (w systemie szesnastkowym).

### Interpretacja

Ten etap zaczyna się w chwili wykonywania

komendy RUN i kończy się z chwilą zakończenia wykonania programu.

Po uruchomieniu programu uaktywnia się procedura, którą w dalszym ciągu będziemy nazywać pętlą interpretera, pobierającą i interpretującą zawartość kolejnych „komórek”. Pętla interpretera rozpoznaje np. czy pobrany kod jest instrukcją, separatorem, końcem linii itp. i wywołuje, w zależności od wyniku tego rozpoznania, odpowiednią procedurę wykonującą zakodowane zadanie. Po wykonaniu tej procedury interpreter wraca do pętli.

W naszym przykładzie pętla interpretera pobiera kod litery a i sprawdza czy jest to kod jakiejś instrukcji. Ponieważ 41 nie jest kodem ani instrukcji ani separatora w interpreterze CMB V2.0 (lub CMB V7.0), to wywoływana jest procedura obsługująca instrukcję podstawienia (LET), która przejmie dalszą interpretację tego zapisu.

Po wykonaniu podstawienia pętla interpretera pobierze kod 3a (jest to kod separatora :). Gdyby w tym miejscu zabrakło dwukropka, to pętla interpretera odesłałaby do procedury obsługującej błędy (jak wiemy w tym miejscu program zakończyłby się komunikatem SYNTAX ERROR). Separator odsyła do pobrania zawartości następnej „komórki”. W naszym przypadku będzie to kod 99 instrukcji PRINT.

Pętla interpretera kończy interpretację linii po napotkaniu kodu 00. W rozważanym przykładzie, po napotkaniu tego kodu i stwierdzeniu, że jest to ostatnia linia programu, interpreter zakończy wykonywanie programu.

Rozbudowa interpretera powinna zatem zawierać elementy, które umożliwią rozpoznanie słów kluczowych (instrukcji), ich kodowanie i rozszyfrowywanie na użytek pętli interpretera i wyprowadzania danych i treści programu z pamięci wewnętrznej.

Warsaw BASIC jest rozbudowanym interpreterem CBM V2.0 we wszystkich wymienionych kierunkach. Szczegółowa realizacja tak rozumianej rozbudowy nie jest oczywiście rzeczą prostą. Nie zważając na to, chcemy zapoznać Czytelników BAJTKA z wybranymi elementami Warsaw BASIC'a, niespotykanymi w innych interpreterach instalowanych w mikrokomputerach domowych i osobistych. Aby można było to uczynić w możliwie prosty sposób, omawiając w kolejnych odcinkach nie tylko zasady działania, ale i pokazując na przykładach działanie tego interpretera, jesteśmy zmuszeni do zawężenia modyfikacji interpretera do etapu interpretacji.

Mamy nadzieję, że zawężenie to pozwoli na zaprezentowanie w następnych odcinkach takiej modyfikacji interpretera CBM V2.0, która pokaże na czym polega dynamiczny podział pamięci wewnętrznej komputera, programowanie proceduralne i strukturalne, tworzenie biblioteki procedur itp. Aby to uczynić należy zacząć już w następnym odcinku od nauczenia pętli interpretera rozpoznawania nowych instrukcji.

*K. Gajewski  
B. Radziszewski*

## CZWARTE WCIELENIE

# DINOZAURA

**Niewiele jest na świecie komputerów ośmiobitowych, które przeżywają drugą młodość w postaci nowych wersji, natomiast na pewno istnieje tylko jeden, który pojawia się na rynku w swoim czwartym wcieleniu. Komputerem tym jest Commodore 64. W drugiej wersji zainstalowano go w innej, większej maszynie (C-128), w trzeciej — zmieniono mu wystrój zewnętrzny (C-64C), w czwartej zmieniono wnętrze.**

Najnowsza wersja jest stosunkowo niedroga, a przy tym, dzięki wprowadzonym zmianom, bardziej niezawodna. Obudowa komputera pochodzi z C-64 natomiast klawiatura z wersji C-64C. Wnętrze komputera zostało całkowicie przeprojektowane i zmienione, a jedyną pozostałością są tu układy interface'ów CIA — 6526 i pamięć ROM oznaczona jako 901225.

System operacyjny oraz interpreter języka BASIC umieszczono w jednym układzie pamięci ROM oznaczonym jako 251913 co znalazło zastosowanie już wcześniej w Commodore 128. Rozkład wyprowadzeń tego układu jest taki sam jak układ wyprowadzeń popularnej pamięci EPROM 27128. Procesor 6510 został tu zastąpiony układem 8500 R4, którego lista rozkątów jest taka sama jak w poprzedniku.

Zupełną nowością jest zastosowanie nowego układu oznaczonego jako 251715 w którym zawarte jest całe 64 KB RAM. Jako ciekawostkę warto przytoczyć tu fakt, że układ ten może być zastąpiony układem o większej (np. 256 KB) pojemności co pozwala wykorzystać ją jako RAM-disk lub też jako normalną pamięć operacyjną dostępną dla procesora w postaci banków po 64 KB każdy.

Wymieniony został także procesor graficzny i dźwiękowy — układy VIC 6569 oraz SID 6581. Nowy procesor graficzny (odpowiedzialny zarówno za tekst jak i grafikę) nosi oznaczenie 8565, układ dźwiękowy natomiast 8580. Firma Commodore tłumaczy wymianę tych układów tym, że do ich zasilania wymagane były dwa napięcia — 12V oraz 5V prądu stałego. Problemem było tu napięcie +12V wytwarzane z napięcia 9V prądu zmiennego pobieranego z regulatora napięcia

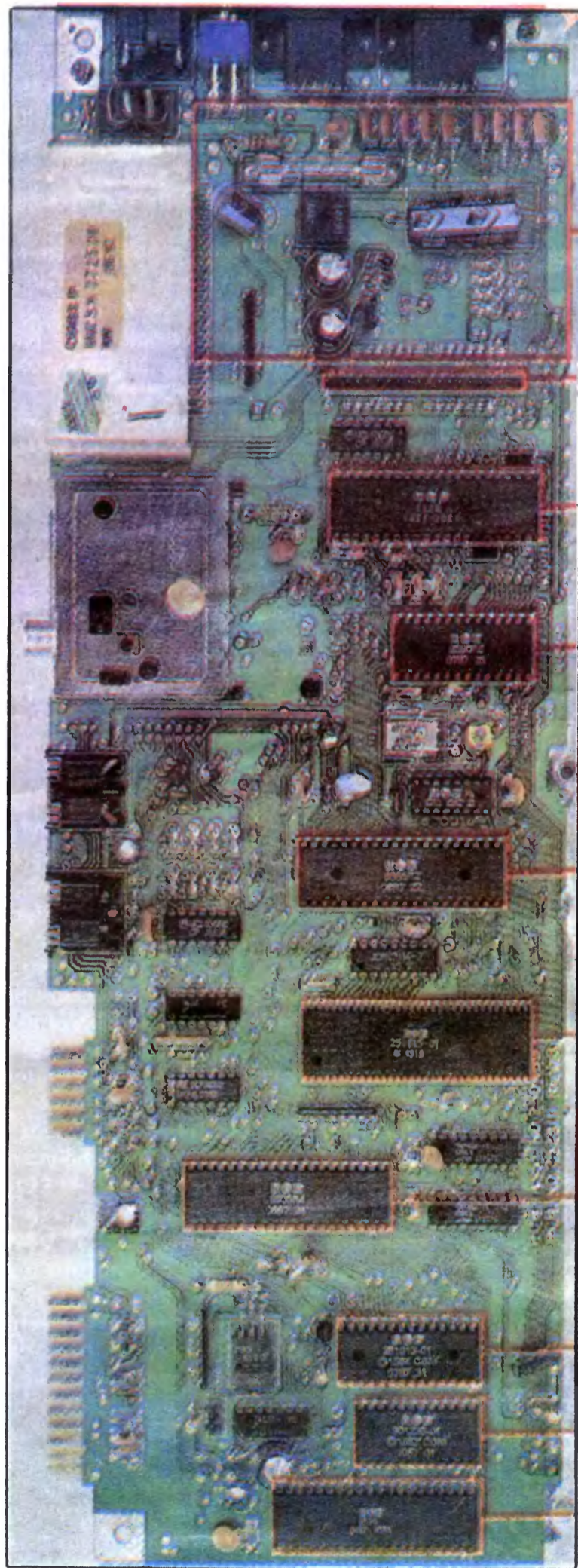
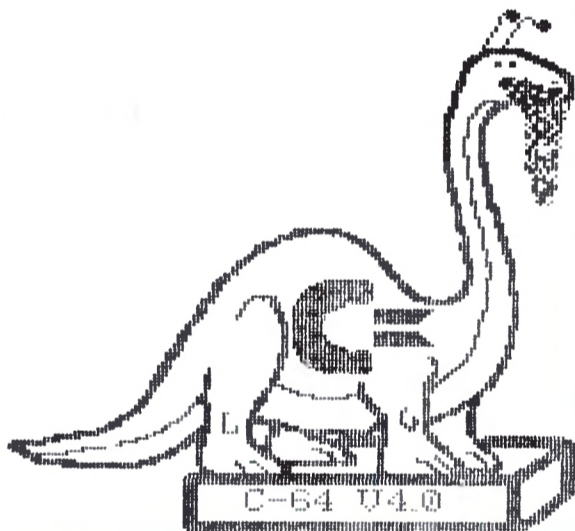
umieszczonego w samym komputerze. Pomijając nawet fakt niezbyt stabilnego napięcia zasilającego delikatne układy, prowadziło to także do wytwarzania się dużych ilości ciepła. Problem ten rozwiązano w elegancki sposób — nawet układy są zasilane jedynie za pomocą 5V. Dodatkową zaletą takiego rozwiązania jest także obniżenie kosztów produkcji.

Kompatybilność nowej wersji C-64 ze starszymi wersjami jest stuprocentowa — dotyczy to również kompatybilności z rozszerzeniami przyłączonymi do portu rozszerzania (EXPANSION PORT) takimi jak FINAL CARTRIDGE, POWER CARTRIDGE, TURBO PROCESS, ACTION REPLAY itp.

Na koniec najważniejsza informacja, jego cena wynosi w RFN od 248—328 DM, (wrzesień/paźdź. 87).

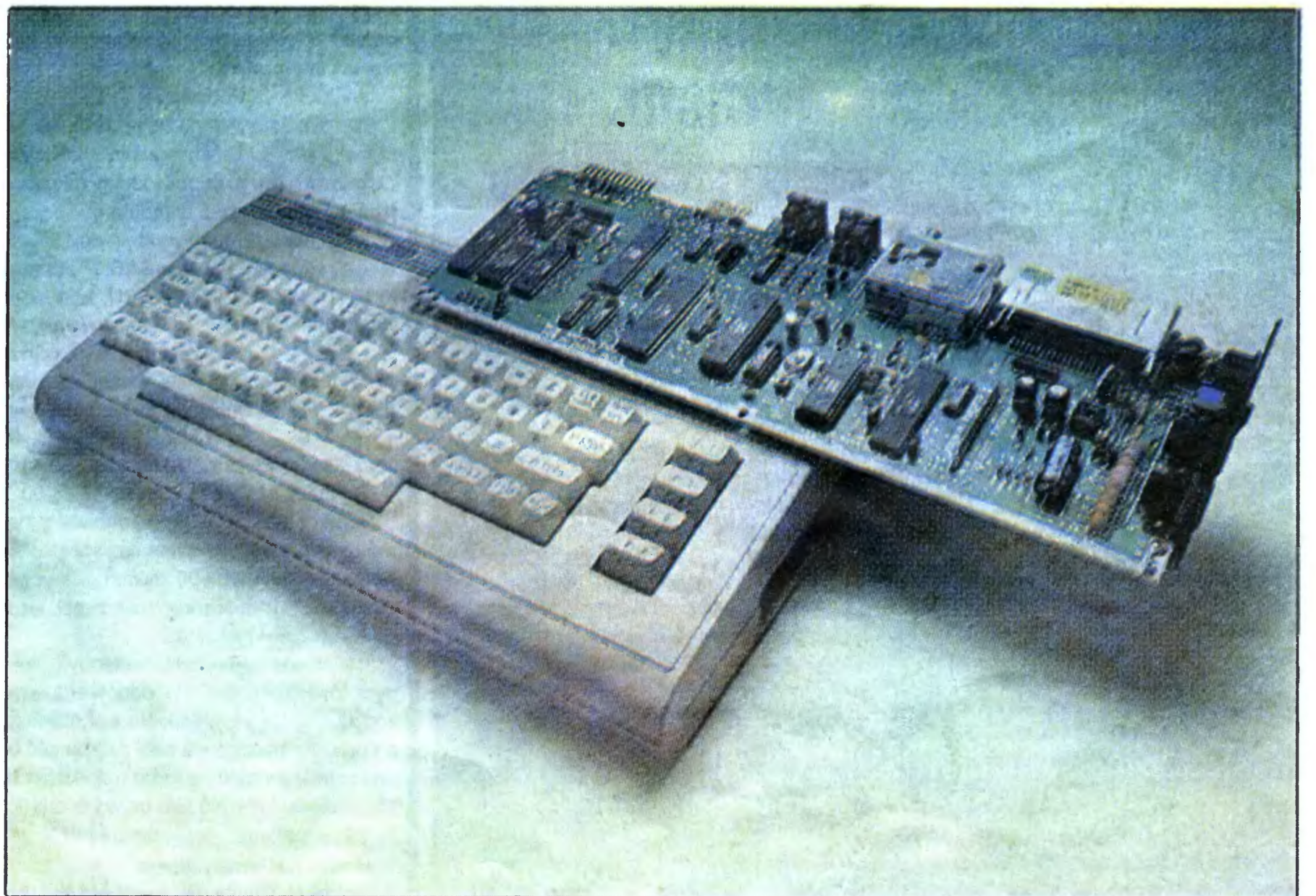
Na podstawie „64'er” 09/87

Jan Jasiński  
COMMODORE CLAN „KOMODA”



Rys 2. PŁYTA GŁÓWNA NOWEGO C-64

- a) moduł zasilania
- b) złącze klawiatury
- c) układ 6526 obsługujący klawiaturę
- d) nowy procesor dźwiękowy
- e) nowy procesor graficzny
- f) pamięć RAM
- g) mikroprocesor
- h) interpreter języka BASIC i system operacyjny
- i) pamięć ROM generatora znaków
- j) układ 6526 obsługujący port użytkownika i szynę szeregową (USER PORT i SERIAL PORT)



Rys 1. CZWARTE WCIELENIE „DINOZAURA” C-64

# ZEPSUŁEM PROGRAM NA TAŚMIE!

**Sytuacja wydaje się beznadziejna. Zepsułem początek programu nagranych na taśmie. Nie mam listingu i w ten sposób wiele godzin spędzonych przy pisaniu programu poszło na marne (nie licząc nabytego doświadczenia). Zepsuty jest tak nagłówek jak i sygnał pilotujący wraz z początkiem programu.**

Była to chwila nieuwagi lub wyłączenie prądu w czasie nagrywania na taśmę w tym samym miejscu, gdzie była poprzednia wersja tego programu. Ale przecież fizycznie reszta programu na taśmie istnieje? Czy nie ma żadnej możliwości aby się do niej dostać? Czy mam wszystko pisać od początku?

Otóż nie, jeżeli znasz dobrze swój komputer i sprawa jest warta tego, aby poświęcić jej więcej czasu. Na początku trochę wiadomości ogólnych o sposobie nagrywania na taśmę przez ZX-Spectrum. Procedura ładująca i weryfikująca dowolny moduł z taśmy znajduje się w ROM-ie pod adresem #0556. Przed wywołaniem tej procedury należy umieścić w rejestrze IX adres początku wgrzanego bloku, w DE jego długość a w akumulatorze typ bloku (0-nagłówek, 255-właściwy blok). Brak ustawienia znacznika C oznacza weryfikację. Nagrana na taśmę ilość bajtów jest zawsze o 2 bajty większa od tej długości, jaką deklarujemy w rejestrach DE. Na samym początku zapisywana jest zawartość rejestru A, co umożliwia sprawdzenie rodzaju bloku, oraz na samym końcu bajt parzystości budowany na bieżąco jako funkcja XOR z wszystkich nagrywanych bajtów. Pozwala to na końcowe sprawdzenie poprawności wczytanego bloku.

W trakcie ładowania modułu do komputera jednym sprawdzianem poprawności ładowania jest odliczanie taktów zegara między kolejnymi zboczami napięcia przychodzącymi do portu #FE. Różnica czasu przekraczająca około 2400 taktów powoduje znany nam wszystkim komunikat „Tape loading error” już w trakcie wczytywania. Nie istnieje żadne inne sprawdzanie, czy aktualna para zboczy reprezentująca „0” lub „1” należy jeszcze do poprzedniego, czy już do następnego bajtu. Po odliczeniu 8 bitów komputer oczekuje następnej ósemki, itd. aż do osiągnięcia żądanej długości bloku.

Kłopot w tym, że opisana wcześniej procedura ładująca wymaga na początku sygnału pilotującego o większej odległości między zboczami oraz specjalnego impulsu synchronicznego, przed właściwym czytaniem bajtów. A przecież my właśnie to zniszczyliśmy! Może więc ominąć część tej procedury i przejść od razu do czytania poszczególnych bitów? Ale nie mamy wtedy żadnej pewności czy nasze czytanie rozpocznie się od początku bajtu! I jest to

prawda. Z tym problemem postaramy się, przynajmniej częściowo, uporać później.

Na razie krótki program, napisany w assemblerze, umożliwiający wykorzystanie procedury ładującej z ominięciem opisanych wcześniej kłopotów.

```
ADRES EQU adres ; tu wstawić początek bloku
ORG 65000
LD IX, ADRES ; ustawienie adresu początku bloku w IX
LD DE, #FFFF ; maksymalna długość (i tak jej nie znamy)
LD C, #06 ; ustawienie koloru ramki
XOR A ; ustawienie znacznika Z, nie będzie sprawdzana zgodność pierwszego bajtu

SCF ; ustawienie znacznika C, ładowanie
EX AF,AF' ; przechowanie znaczników w rejestrze F'
DI ; wyłączenie przerwań maskowalnych
LD HL,POWRÓT ; odłożenie na stosie adresu powrotnego
PUSH HL ; z procedury ładującej
JP #05A3 ; skok do czytania pierwszego bitu

POWRÓT LD HL, #FFFF ; obliczanie ilości wczytanych bajtów

SBC HL,DE ; załadowanie ilości wczytanych bajtów
LD B,H ; rejestrów BC
LD C,L ; włączenie przerwań maskowalnych
EI

RET
```

Przykładowy loader w ZX-Basic dla tej procedury przedstawia się następująco:

```
10 CLEAR 29999: LET adres =30000
20 FOR n=0 TO 28: READ a: POKE 65500+n, a: NEXT n: STOP
100 DATA 221, 33, adres-256*INT (adres/256), INT (adres/256)
110 DATA 17, 255, 255, 14, 6, 175, 55, 8, 243, 33, 240, 255, 229, 195, 163, 5
120 DATA 33, 255, 255, 237, 82, 68, 77, 251, 201
```

Po zainstalowaniu procedury ustawiamy magnetofon w miejscu gdzie jesteśmy pewni, że znajduje się już nasz odzyskiwany kawałek programu. Następnie włączamy magnetofon i wysyłamy w trybie bezpośrednim PRINT USR 65500. Istotna jest tutaj kolejność — najpierw włączamy magnetofon a później uruchamiamy procedurę. Wynika to z faktu, że procedura natychmiast przechodzi do czytania kolejnych bitów, co w przypadku niewłączonego jeszcze magnetofonu spowoduje powrót do ZX-BASIC. Istnieje także możliwość złego trafienia w przedziały czasowe nagrane na taśmę, co także zakończy się natychmiastowym powrotem do interpretera. Musimy wtedy przewinąć taśmę i ponownie uruchomić procedurę.

Po pozytywnym zakończeniu wgrzywania na ekranie zostanie wprowadzona ilość wczytanych do komputera bajtów.

Powróćmy teraz do problemu kolejności bitów w poszczególnych bajtach. Sytuacja jest najprostszą w przypadku programu napisanego w ZX-BASIC i nagranych bez wartości zmiennych. Wtedy znamy ostatni bajt (przed bajtem kontrolnym) jaki został nagrany na taśmę — jest nim liczba 13 (ENTER), która kończy każdy wiersz programu. Posłużmy się przykładem. Niech dwa ostatnie bajty wczytane z taśmy wynoszą 48 i 107. Rozkład bitów w tych bajtach oraz w liczbie 13 jest następujący (zaczernienie oznacza jedynekę)

```

      48      107      13
      _____
      00110000 01100111 00001101

```

Liniją przerywaną zaznaczono lokalizację bajtu o wartości 13 w dwóch ostatnich wczytanych bajtach. Wynika z tego, że ostatni bajt powinien być przesunięty o trzy bity w prawo oraz trzy bity najbardziej znaczące winny być wzięte z poprzedniego bajtu. Funkcję przesunięcia o trzy bity możemy zrealizować dzieląc dany bajt przez 8. Aby przesunąć wszystkie bajty odzyskiwanego programu wystarczy napisać krótki program:

```
1000 LET końcówka=0: FOR n=adres TO adreskonca
1010 LET bity przesuwane=INT (PEEK n/8): LET nowakońcówka=PEEK n-8*bity przesuwane
1020 POKE n, 32*końcówka+bity przesuwane: LET końcówka=newakońcówka: NEXT n
```

Musimy jeszcze ustalić prawidłowy początek naszego bloku. W tym celu przeglądamy początek wczytanego modułu zwracając uwagę, gdzie znajduje się liczba 13. Po niej powinna następować dwubajtowa liczba opisująca numer wiersza programu. Liczba 13 może jeszcze występować wewnątrz 5-cio bajtowej reprezentacji liczby w ZX-BASIC, po liczbie 14. Gdy zlokalizowaliśmy prawidłowy początek wiersza nagrywamy tak odzyskany moduł na taśmę jako moduł binarny instrukcją „SAVE”nazwa”CODE adrespoczątku, adreskońca-adrespoczątku+1. Pozostaje nam jeszcze tylko wgrać go ponownie pod programem COPY-COPY. Zmieniamy informacje zawarte w nagłówku:

1. wpisujemy do pierwszego bajtu 0,
2. wpisujemy do bajtu 14-tego wartość 32768,
3. wpisujemy do bajtu 16-tego długość modułu (wielkość podawana jako „dług” w trakcie wyświetlania nagłówków).

Po ponownym nagraniu na taśmę dysponujemy już odzyskanym fragmentem naszego „straconego” programu.

Niestety, w przypadku pozostałych modułów (tablice, binaria i programy w ZX-BASIC nagranych wraz ze zmiennymi) nie mamy żadnej informacji o ostatnim bajcie modułu. Pozostaje więc 7 możliwości przesuwania bitów w kolejnych bajtach, która z nich jest prawidłowa, możemy tylko próbować ocenić przyglądając zawartość pamięci po kolejnych przesunięciach kierując się własnym doświadczeniem i wiadomościami o tym, co powinien zawierać dany moduł. Oczywiście jest to wtedy bardzo pracochłonne i należy się zastanowić czy naprawdę warto odzyskiwać dany moduł o i tak zniszczonym bezpowrotnie początku.

*Wojciech Indyk*

## KOMPUTER I TELEFON

Już od 1979 roku funkcjonuje we Frankfurcie nietypowy system telefonicznej informacji kolejowej. Zamiast ludzi zdecydowano się zatrudnić komputer. Jest on niezawodnym pracownikiem, który nigdy nie męczy się i zawsze jednakowo uprzejmie traktuje

klientów. Podróżni zdążyli się już przyzwyczaić do jego monotonnego głosu i nadali mu zdrobniałe imię „Karolek”.

Komputer udziela informacji 24 godziny na dobę. Prowadząc dialog pozdrawia on najpierw grzecznie pasażera, a następnie prosi o wykręcenie

przy pomocy tarczy impulsowej telefonu kodu pocztowego miejscowości docelowej, daty podróży oraz przybliżonej godziny odjazdu. Karolek jest w stanie poinformować o połączeniach z 300 miejscami w całej Europie. Wszystkie niezbędne do tego dane są prze-

chowywane w jego elektronicznej pamięci. System ten zyskał sobie duże uznanie u pasażerów.

Pomysł zastosowania tarczy telefonicznej jako urządzenia wspomagającego dialog człowieka z komputerem wykorzystano również w systemie przyjmowania zamówień. W tym przypadku należy wykręcić katalogowy numer towaru, który pragniemy zakupić. (jj)

# OD ŚRODKA CZ. 2

**Znacie już cały podział pamięci ZX-Spectrum „mamy nadzieję, że zapamiętaliście coś z poprzedniego artykułu). A więc do dzieła. Dziś poznamy sposób zapisu programów na taśmie i zaczniemy się włamywać do programów napisanych w BASIC-u (pamiętaj, że praktycznie każdy program ma chociażby procedurę ładującą napisaną w BASIC-u, a jeśli jej nie posiada, to w zasadzie wcale nie jest zabezpieczony).**

Na razie weźmy się jednak do sposobu zapisu programów na taśmie — przypomnij sobie co widać i słychać podczas wczytywania jakiegokolwiek programu.

Przez ok. 1 — 2 sekundy, na obrzeżu ekranu widoczne są szerokie, czerwono-niebieskie pasy oraz słychać ciągły dźwięk. Jest to tzw. pilot, który umożliwia komputerowi zsynchronizowanie się z sygnałem z taśmy, który ma odbierać. Potem przez krótką chwilę pojawiają się cienkie, migające, żółto-granatowe paski, oznaczające, że komputer wczytuje do pamięci informacje. Jest ich 17 bajtów — tzw. nagłówek. Pojawia się napis „Bytes:” „Program:”, „Character array:” lub „Number array:”, a potem po chwili przerwy zaczyna się drugi pilot (nieco krótszy), a po nim wczytuje się właściwy program. Zajmiemy się teraz nagłówkami, gdyż zawarte są w nich najważniejsze informacje o wczytywanych programach.

Nagłówek — jak już wspomnieliśmy — składa się z 17 bajtów. Będziemy je numerować od 0 do 16 (patrz rys.1). Zerowy bajt oznacza typ bloku. Jest on równy 0 — jeśli jest to program w BASIC-u, 3 — jeśli blok kodu maszynowego (nagrany przez SAVE „...”**CODE**... albo **SAVE** „...”**SCREENS**, które oznacza dokładnie to samo, co **SAVE** „...”**CODE 16384,6912**). Jeśli natomiast nagłówek ten poprzedza zbiór będący tablicą zmiennych BASIC-a (nagrany przez SAVE „...” DATA...), to jest on równy 1 — dla tablic liczbowych lub 2 — dla tablic znakowych.

Następne 10 bajtów, to nazwa wczytywanego bloku, czyli tekst pojawiający się po załadowaniu nagłówka, za napisem „Program:”, „Bytes:” czy innym.

Bajty 11. i 12. zawierają dwubajtową liczbę (pierwszy bajt mniej znaczący) oznaczającą długość bloku danych, którego ten nagłówek dotyczy.

W zależności od typu wczytywanego bloku, bajty od trzynastego do szesnastego są w różny sposób interpretowane.

Zacznijmy od nagłówków programów napisanych w BASIC-u. Bajty 13. i 14. oznaczają numer linii, od której ma się uruchomić program, jeśli był nagrany przez **SAVE** „...”**LINE** nr. Jeśli program był nagrany bez opcji **LINE** czyli po wczytaniu nie uruchomi się samoczynnie, to wartość tej liczby jest większa od 32767. Jednym ze sposobów unieszkodliwiania zabezpieczeń samostartujących programów w BASIC-u jest zmiana tych dwóch bajtów w nagłówku na wartość większą od 32767.

Bajty 15. i 16. zawierają liczbę oznaczającą długość samego programu w BASIC-u (bo **SAVE** „...” lub **SAVE** „...”**LINE** nagrywają program wraz ze wszystkimi zmiennymi, tzn. zawartość pamięci od bajtu wskazywanego przez zmienną systemową **PROG**, do bajtu określonego zmienną **E\_LINE**). Gdy od całkowitej długości bloku (bajty 11. i 12.) odejmiemy tę liczbę, dowiemy się ile bajtów w tym bloku zajmują zmienne BASIC-a.

To wszystko, jeśli chodzi o nagłówki programów w BASIC-u.

W nagłówkach kodu maszynowego

(„Bytes:”) bajty 15. i 16. nie są wykorzystywane, natomiast 13. i 14. tworzą dwubajtową liczbę określającą pod jaki adres należy wczytać następujący po nagłówku blok.

W nagłówkach tablic wykorzystywany jest spośród tych czterech, tylko bajt 14., który przedstawia nazwę wczytywanej tablicy. Jest on zapisany tak, jak nazwy wszystkich zmiennych BASIC-a (w obszarze od **VARS** do **E\_LINE**), tzn. trzy najstarsze bity oznaczają typ zmiennej (tutaj będzie to tablica liczbowa lub znakowa) a pięć mniej znaczących bitów — nazwę tej zmiennej\*).

Na listingu 1. przedstawiony jest program tworzący procedurę umożliwiającą odczytywanie nagłówków programów zapisanych na taśmie. Po jego dokładnym wpisaniu (uważaj na właściwą ilość spacji w liniach z instrukcjami **DATA!**) uruchom go przez **RUN** i poczekaj chwilę. Na ekranie pojawi się informacja o adresach początku i końca procedury oraz o jej długości, która powinna wynosić 284 bajty. Jeśli tak nie jest, sprawdź, czy na pewno wpisałeś wszystkie linie programu. Jeśli wszystkie dane były poprawne, to na twojej taśmie znajdzie się procedura „czytacz”, dzięki której będziesz mógł odczytać każdy nagłówek. Po nagraniu jej na taśmę możesz już skasować program z listingu 1.

Procedura „czytacz” jest relokowana, tzn. będzie działała poprawnie niezależnie od tego, pod jakim adresem w pamięci się znajdzie. Możesz ją więc wczytać przez:

**LOAD „czytacz”CODE adres**  
a następnie uruchamiać przez  
**RANDOMIZEUSR adres**

i to nawet wielokrotnie. Po uruchomieniu, procedura wczytuje pod adres 23296 (do bufora drukarki) pierwszy napotkany nagłówek. Jeżeli ze względu na podłączone urządzenia zewnętrzne nie odpowiada ci ten adres, to możesz go zmienić zastępując w linii 200 programu z listingu 1. liczbę 005B heksadecymalnym adresem, pod który chciałbyś wczytywać nagłówki (dwie pierwsze cyfry są młodszym bajtem tego adresu). Aby po takiej zmianie uniknąć sprawdzania sumy kontrolnej w tej linii, na końcu tekstu

ujętego w cudzysłowy zamiast spacji i czterech cyfr sumy kontrolnej umieść literę „s” poprzedzoną czterema spacjami.

Po wczytaniu nagłówka procedura odczytuje informacje w nim zawarte i wraca do BASIC-a, lecz wczytanego nagłówka nie niszczy, więc jeśli chcesz, możesz go dodatkowo obejrzeć wykorzystując funkcję **PEEK**.

Samo odczytywanie nagłówków to jednak trochę mało, by móc włamywać się do bloków zapisanych na taśmie. Trzeba jeszcze wiedzieć, co z tymi blokami zrobić, aby umieścić je w pamięci, nie pozwalając im się przy tym uruchomić.

W przypadku bloków typu „Bytes” wystarczy zwykle załadować je pod wymuszony adres, powyżej **RAMTOP**-u (tzn. powyżej komórki pamięci wskazywanej przez zmienną systemową **RAMTOP**), np. przez:

**CLEAR 29999: LOAD ""CODE 30000**

Metoda ta skutkuje, jeśli tylko wczytywany blok nie jest zbyt długi (może mieć maksymalnie ok. 40K). Dłuższe bloki mogą się po prostu nie zmieścić w pamięci — konieczne jest wówczas podzielenie ich na kilka krótszych części. Niedługo dowiemy się jak to robić.

Podobnie, w przypadku tablic, załadowanie ich nie nastęrcza raczej kłopotów — wystarczy użyć normalnej w takich wypadkach instrukcji **LOAD ""DATA...**

Gorzej natomiast wygląda wczytywanie programów w BASIC-u. Są one zwykle nagrywane przez **SAVE** „...”**LINE**, a na samym początku linii, od której mają się uruchomić umieszczone są instrukcje zabezpieczające je przed zatrzymaniem. Najprostszym rozwiązaniem jest ładowanie programu nie przez **LOAD** „...”, lecz **MERGE** „...”, jednak metoda ta nie zawsze daje rezultaty. Z tej beznadziejnej wówczas sytuacji istnieją dwa zasadnicze wyjścia: podmienić nagłówek programu lub użyć przedstawionego niżej programu „load/merge”.

Pierwsze z nich polega na zastąpieniu nagranych na taśmie nagłówka przez taki sam, ale nie powodujący uruchomienia programu. Można w

0	Typ	Typ bloku :
1		<div style="border: 1px solid black; padding: 5px;">                     0 - Program                      1 - Number array                      2 - Character array                      3 - Bytes                 </div>
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12	Długość	długość bloku danych - ile bajtów trzeba będzie zaraz wczytać z taśmy
13	Start	Bytes: adres ładowania bloku Program: numer linii od której program się uruchomi Array: bajt 14-ty - nazwa zmiennej, 13-ty nieważny
15	Program	Program: długość samego BASIC-a, ( bez zmiennych ) Bytes,Array: nieważne
16		

tym celu użyć programu „COPY COPY” — wczytać nagłówek programu, wcisnąć BREAK, a następnie instrukcją LET zmienić jego parametr START na liczbę większą od 32767 (tzn. wykonać np. **LET 1=„32768** — jeśli przerabiany nagłówek był wczytany jako pierwszy zbiór). Tak zmodyfikowany nagłówek nagrywamy gdzieś na taśmę. Usuwamy z pamięci program COPY COPY i wpisujemy LOAD ””. Wczytujemy stworzony przed chwilą nagłówek, lecz zaraz po jego zakończeniu zatrzymujemy taśmę. Teraz do magnetofonu wkładamy kasetę z programem, do którego się wlamujemy, ale ustawioną zaraz za nagłówkiem tego programu — tak, aby wczytać tylko treść programu, bez oryginalnego nagłówka.

Druga metoda jest o wiele wygodniejsza:

Wprowadzamy do pamięci (z klawiatury lub z taśmy) program „load/merge” zamieszczony na listingu 2. Po uruchomieniu zaczyna on czekać na pierwszy program w BASIC-u, który znajduje się na taśmie. Wczytuje go zupełnie tak samo, jak instrukcja LOAD ””, lecz po załadowaniu nie pozwala na jego uruchomienie — drukuje jedynie komunikat „0 OK”, z informacją od której linii wczytany program miał się uruchomić.

Na listingu 3. przedstawiona jest procedura, którą tworzy ten program poprzez POKE-i. Zasadniczą jej instrukcją jest CALL 1821. Począwszy od adresu 1541, w ROM-ie znajduje się procedura interpretująca instrukcje **SAVE, LOAD, VERIFY** i **MERGE**. **CALL 1821** wskazuje do środka tej procedury. Początkową jej część (odczytującą parametry tych instrukcji) pomijamy, zastępując to w liniach 40 — 90 naszej procedury: Najpierw przez RST 48 rezerwujemy 34 bajty w obszarze WORKSP\*\*), adres wolnego miejsca przenosimy z rejestru DE do IX, w linii 80 zaznaczamy w zmiennej systemowej T\_ADDR, że chodzi nam o **LOAD**, a nie np. **VERIFY**, a w pierwszym bajcie nagłówka do porównywania umieszczamy wartość 255, oznaczającą, że ma być wczytany pierwszy napotkany blok (tzn. program w BASIC-u). Reszta dokonuje procedurę z ROM-u, działając identycznie jak **LOAD** ””. Różnica w działaniu LOAD a „load/merge” polega na tym, że po całkowitym wczytaniu programu, zamiast wrócić przez **RET** do interpretera BASIC-a, przepisujemy zawartość zmiennej NEWPCC („numer linii do której ma być skok”) do PPC („numer ostatnio wykonywanej linii”) oraz nie dopuszczamy do uruchomienia programu — wykonujemy RST 8 z komunikatem „0 OK”. Dzięki temu przepisaniu, w wydrukowanym komunikacie znajdzie się informacja o numerze linii, od której wczytany program miał się uruchomić.

Niecierpliwie czekając na kolejny odcinek z tego cyklu poproś odczytywać nagłówki bloków i wczytywać zabezpieczone programy w BASIC-u. Za miesiąc nimi właśnie dokładniej się zajmiemy.

Tomasz Surmacz  
Robert Dudzik

### Listing 1.

```

10 CLEAR 59999: LET pocz=60000
20 LET adr=pocz
30 RESTORE : READ a,b,c,d,e,f
40 DATA 10,11,12,13,14,15
50 LET nr=200: RESTORE nr
60 LET s=0: READ a$: IF a$="." THEN GO TO
130
70 FOR n=1 TO LEN a$-5 STEP 2
80 LET w=16*VAL a$(n)+VAL a$(n+1)
90 POKE adr,w: LET adr=adr+1: LET s=s+w
100 NEXT n
110 IF VAL a$(n TO )<>s THEN PRINT "błąd w
linii ";nr: STOP
120 LET nr=nr+10: GO TO 60
130 PRINT "wszystkie dane Ok." "Początek :
";pocz "koniec : ";adr-1 "długość : ";adr-p
ocz
140 SAVE "czytacz"CODE pocz,adr-pocz
150 REM
200 DATA "21920009E5DD21005BDDE51111 1246"
210 DATA "00AF37CD5605DDE130F23E02CD 1531"
220 DATA "011611C009DD7E00CDOA0CDDE5 1265"
230 DATA "D113010A00CD3C202A7B5CD1E5 1231"
240 DATA "D5ED537B5C010900CD3C20DD46 1346"
250 DATA "0CDD4E0BCD2B2DCDE32DE1DD7E 1664"
260 DATA "00DD460EDD4E0DC5A7202BEB01 1292"
270 DATA "1900CD3C20D5DD4610DD4E0FCD 1361"
280 DATA "2B2DCDE32DD1C178E6C0206EC5 1848"
290 DATA "011300CD3C20C1CD2B2DCDE32D 1280"
300 DATA "185EFE03203601700009EB0111 836"
310 DATA "0018E60D449C75676F9D9B2073 1281"
320 DATA "616D65676F2070726F6772616D 1313"
330 DATA "75200D4175746F737461727420 1161"
340 DATA "2D206C696E696120D2EE1B01B1 1239"
350 DATA "0009EB010900CD3C20C13E1FA0 997"
360 DATA "F660D7DD7E003D28033E24D73E 1383"
370 DATA "28D73E29D7E1227B5C3E0DD7C9 1538"
380 DATA "040B1C2020201C0000101B1030 268"
390 DATA "100C0008103840380478000D41 430"
400 DATA "64726573209C61646F77616E69 1357"
410 DATA "61200D5461626C69636120 862"
420 DATA ". "

```

### Listing 2.

```

1 REM load/merge TS & RD 1987
2 FOR n=60000 TO 60025: READ a: POKE n,a
: NEXT n
3 RANDOMIZE USR 60000
4 DATA 1,34,0,247,213,221,225,253,54,58,
1,221,54,1,255,205,29,7,42,66,92,34,69,
,92,207,255

```

\* Sposób zapisu zmiennych BASIC-a w pamięci jest przedstawiony w dołączonym do Spectrum ang. lub niemieckim podręczniku na stronach 122 — 124, oraz w książce „Przewodnik po ZX-Spectrum” na str. 71.

\*\* Pierwsze 17 bajtów z tych 34 zostaje wypełnione przez procedurę z ROM-u po odczytaniu parametrów wykonywanej instrukcji BASIC-a, drugie 17 to miejsce na wczytywany nagłówek. Obydwa te nagłówki są następnie porównywane, zanim zostanie wczytany blok danych (sprawdzana jest zgodność nazwy i typu bloku).

### Listing 3.

```

10 ; load/merge
20 ;
30 ORG 60000
40 LD BC,34
50 RST 48
60 PUSH DE
70 POP IX
80 LD (IX+58),1
90 LD (IX+1),255
100 CALL 1821
110 LD HL,(23618)
120 LD(23621),HL
130 RST 8
140 DEFB 255

```

## MÓJ MIKRO-KOMPUTER ZX Spectrum



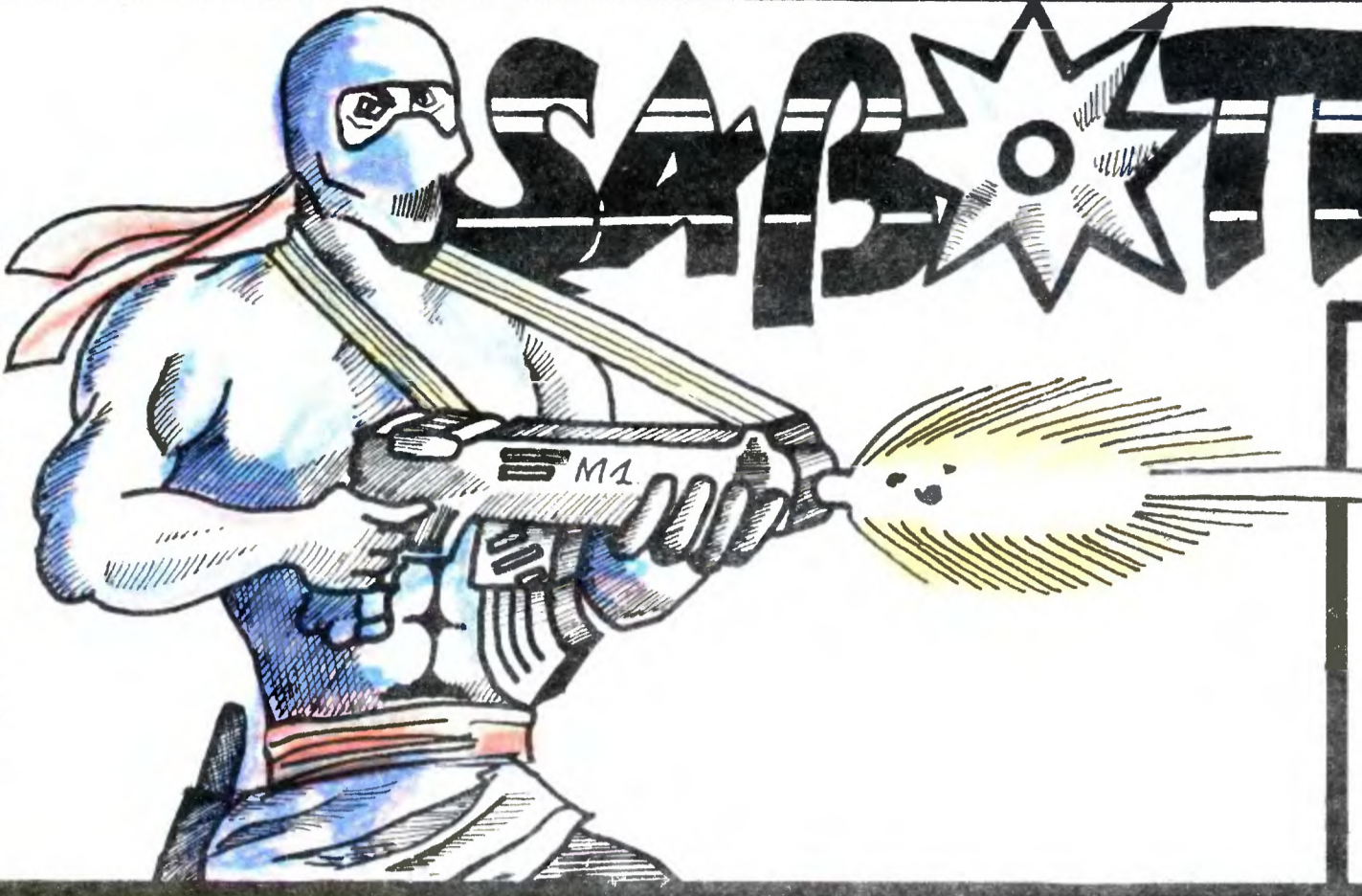
## MÓJ MIKROKOMPUTER ZX SPECTRUM

Minęły już czasy, gdy poszukiwania jakiegokolwiek książki o mikrokomputerach z góry skazane były na niepowodzenie. Szczególnie dużej liczby publikacji doczekało się Spectrum. Kolejną pozycją skierowaną do sympatyków Sinclaira jest „Mój mikrokomputer ZX Spectrum” Rolanda Waclawka, wydaną przez Młodzieżową Agencję Wydawniczą.

Autor koncentruje się wyłącznie na BASIC-u. Opisane zagadnienia w sporej części pokrywają się z fabryczną instrukcją, ale można też znaleźć kilka użytecznych i oryginalnych ciekawostek. Książkę czyta się bardzo przyjemnie, nie nuży ona nadmiarem zbyt mądrych sformułowań. W przypadku literatury popularno-naukowej bardzo ważne jest, by była ona zrozumiała dla jak największej liczby czytelników. I to się autorowi udało. „Mój mikrokomputer ZX Spectrum” jest książką, od której można zacząć edukację. Zawiera informacje najważniejsze, zaprezentowane w formie gwarantującej, że nie zrazimy się do komputera.

(j.j.)  
R. Waclawek, „Mój mikrokomputer ZX Spectrum”, Młodzieżowa Agencja Wydawnicza, Warszawa 1987, wydanie I, nakład 60 tys. egz. Cena 180 zł.

RYŚ: MACIEJ STAWICKI '87



# SABOTEUR II

Jednym z przebojów 1986 roku była gra firmy DURELL zatytułowana SABOTEUR. Autor, Mike Richardson, już w połowie 87 roku wydał drugą część tej gry. Tak, jak pierwszy sabotażysta, miał za zadanie wykraść z warowni dyskietkę, podłożyć bombę i uciec śmigłowcem, tak jego następczyni ma do wykonania misję dużo bardziej skomplikowaną.

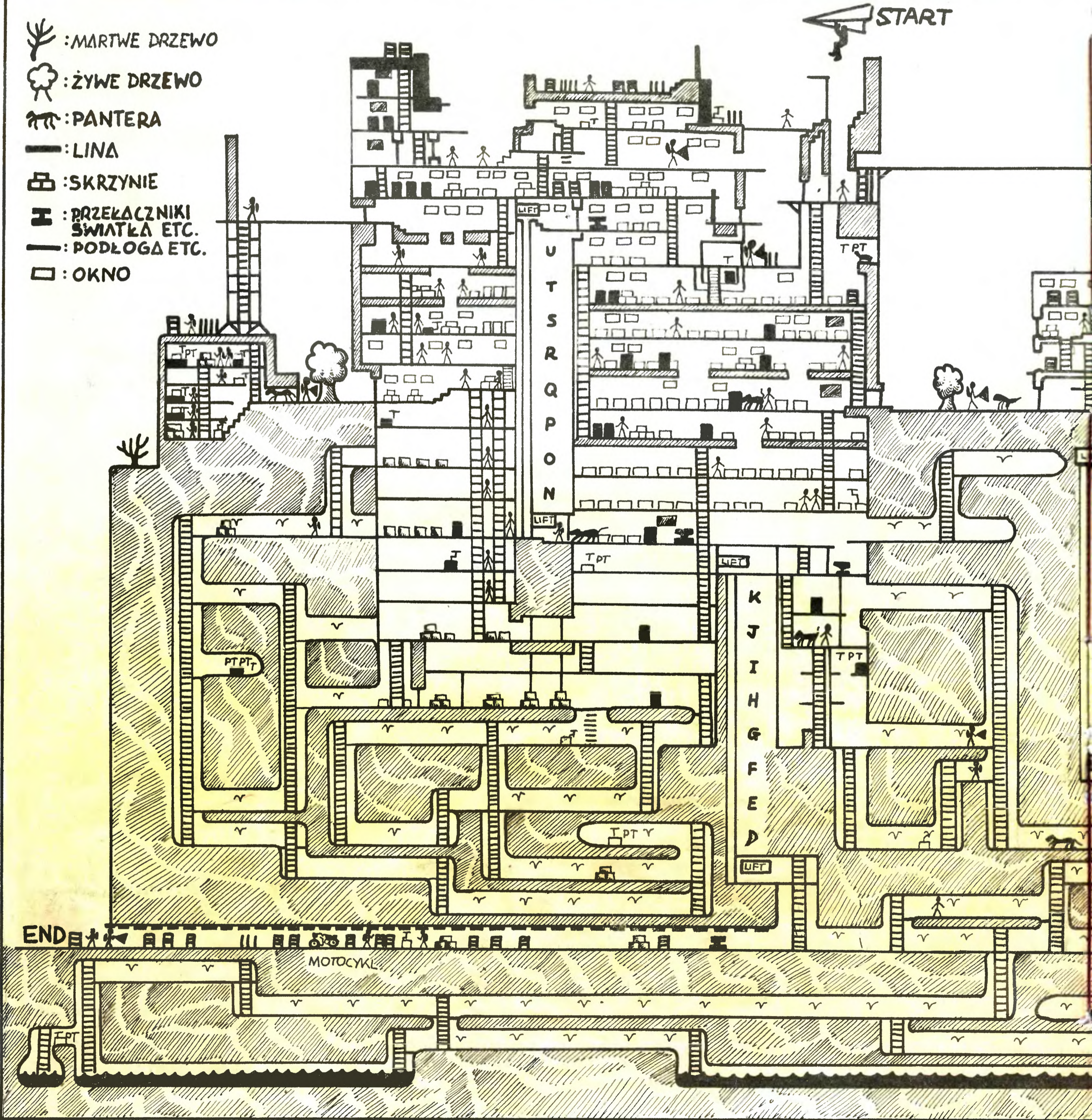
Sabotażystka dostaje się do warowni przy pomocy lotni. W budynku czekają same przyjemności — ogromni strażnicy z laserami (lub bez), pantery, drzewka, windy, telewizory, pudełka, nietoperze, zalane korytarze i wiele innych.

Na wszystko Strażnik odpowie na ziemię bez życia po ciosie można się wdrapać, wolać windy, wydzie, schylić się, urządzić maratony wolnym.

Teraz coś o jest z kilku kondrabinami i... Mamy też linę dwoma słupami, płót elektryczny, Stoi tam mdatny w ucieczce.

Co właściwie haterka? Zależy ziomu trudności nia poziomu w potrzebna jest ukończeniu piermy hasło do drugę kilka pierwszJONIN, 3 — KIMSAIMENJITSU.

-  : MARTWE DRZEWO
-  : ŻYWE DRZEWO
-  : PANTERA
-  : LINA
-  : SKRZYNIĘ
-  : PRZEŁĄCZNIKI ŚWIATŁA ETC.
-  : PODŁOGA ETC.
-  : OKNO





jednak jest sposób. Jedno uderzony pada, pantera kończy w szczękę, na drzewo pać, można też przyjąć z pudełka narzędzi przed nietoperzem i pływacki stylem do

dyнку. Zbudowany gnacji połączonych iurami w podłódze. pociągnięta między rakietę na wyrzutni, dwie windy i podziemie, bardzo przy-

na zrobić nasza bo od wybranego po 1 do 9. Do wybranego niż pierwszy znajomość hasła. Po wszej misji otrzymajęj itd. Dla leni podajęh hasel: misja 2\* — 4 — KUJI KIRI, 5 —

To jeszcze nie koniec urozmaiceń. Pierwsza misja polega na ucieczce motocyklem z budynku, druga — piechotą, w trzeciej trzeba już znaleźć kawałek taśmy perforowanej (ukryta w skrynkach). Potem trzeba będzie wylączyć elektryczny płot i wykonywać wiele innych wymagających sprawnych rąk czynności. Ukończenie dziewiątej misji jest bardzo trudne, trzeba tam wykonać wszystko naraz (płot, 14 części taśmy itp.).

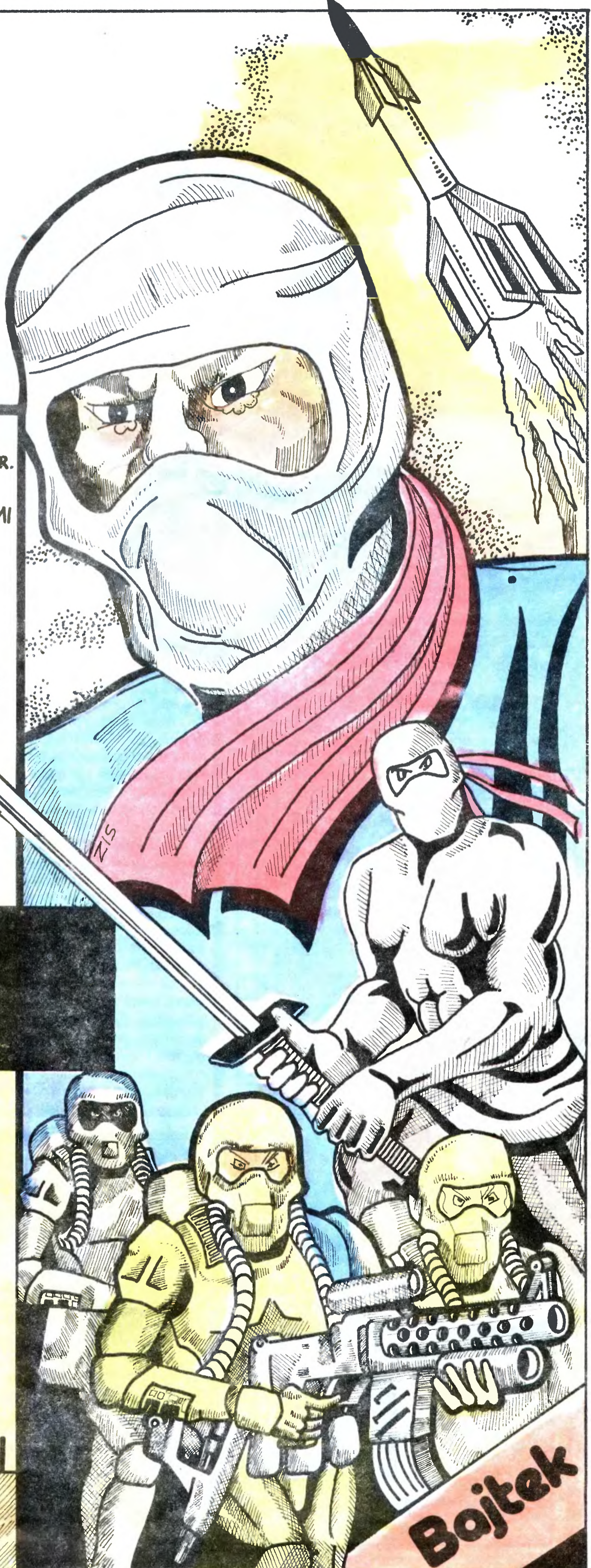
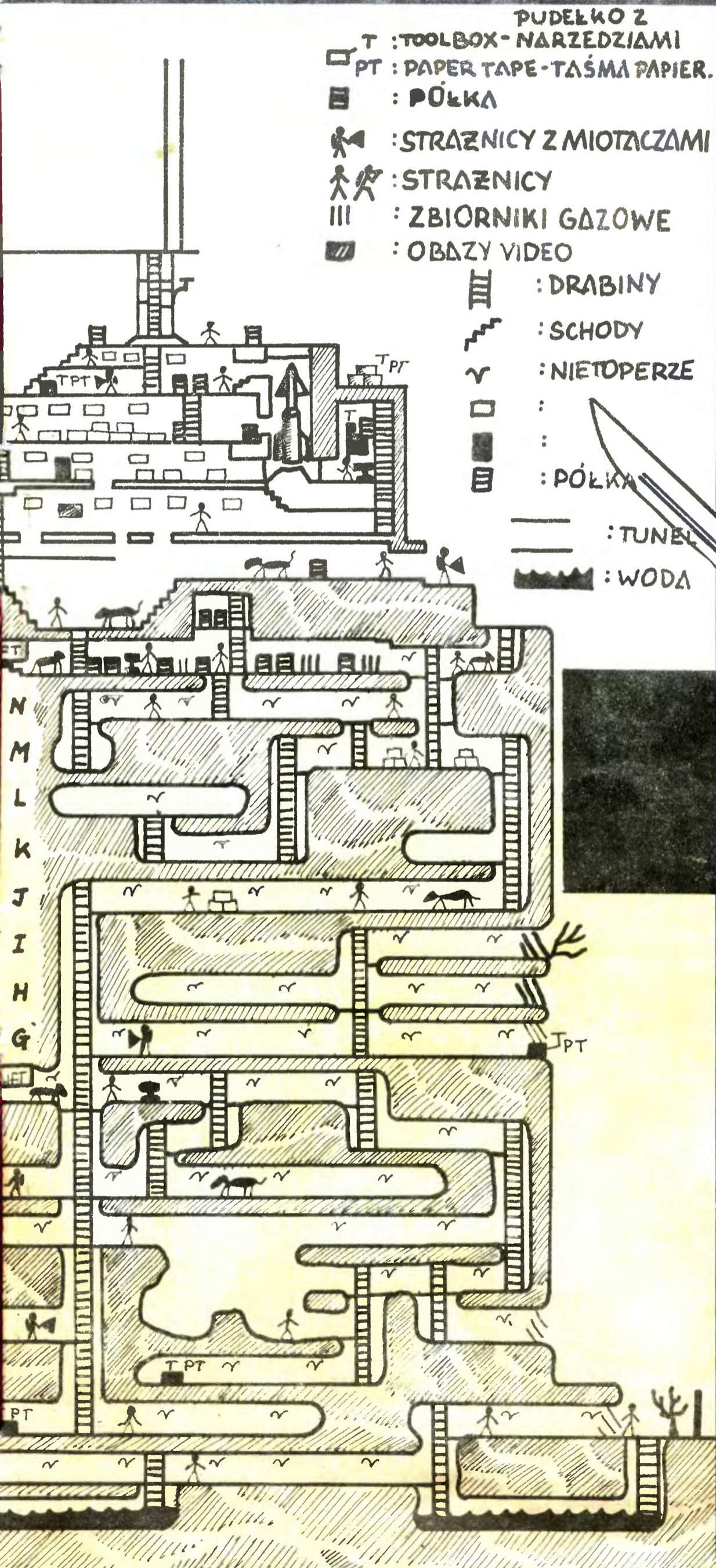
Tak, jak w pierwszej części, na ekranie pokazana jest zawartość kieszeni sabotażystki, ewentualne interesujące przedmioty w pobliżu, stan energii (właściwie siły vitalnej) i zegar. Te dwa ostatnie wskaźniki można wyeliminować, ale o tym opowie PO-KERzysta na następnych stronach.

Autor: Mike Richardson

Firma: DURELL

Komputer: ZX Spectrum 48/+, Commodore 64/128, Amstrad/Schneider

(mp)



**Bajtek**

# 10

BAJTKOWA LISTA PRZEBOJÓW 2/88

Drugie tegoroczne notowanie nie przyniosło rewelacji. Zniknął nawet BOULDER DASH, chyba już bezpowrotnie. Propozycji też jest niewiele — tylko 1911 na 103 tytuły.

	ATARI	AMSTRAD	COMMODORE	SPECTRUM
1 POP EYE ↑		x	x	x
2 BARBARIAN ↑	x	x	x	x
3 NIGHT SHADE ↑		x		x
4 ANTIRIAD ↓		x	x	x
5 EQUINOX !		x	x	x
6 MISS PACMAN ↓	x	x	x	x
7 CHEQUERED FLAG !		x		x
8 GLADIATOR ↓	x	x		x
9 DAM BUSTERS !		x	x	x
10 GREAT ESCAPE ↓		x	x	x

Nagrody — zestawy programów komputerowych otrzymują Joanna Molenda z Krakowa i Janusz Juszczyk z Kalisza.



## KUNG-FU MASTER

**Thomas** wiele sobie obiecywał po pierwszej randce z Sylwią. Długo spacerowali po wąskich uliczkach Szanghaju. Kiedy zapadł zmrok, postanowili wracać do domu. Byli już blisko celu, gdy nagle z ciemnego zaułka wynurzył się podejrzany osobnik. W jego dłoni błysnęło ostrze noża. Reakcja Thomasa była natychmiastowa. Napastnik nie spodziewał się, że Thomas był adeptem szkoły wschodniej sztuki walki w klasztorze Shao-Lin. Dobrze wymierzone kopnięcie wytrąciło nóż z ręki bandyty, a silny cios w podbródek powalił go na ziemię. Nie był to jednak koniec walki. Zewsząd nadbiegały ciemne typki, by pomścić swego kolegę. Thomas zajęty walką nie zauważył, że zamaskowane postacie związały i uprowadziły Sylwię. W oka mgnienia teren opustoszał. Thomas rozejrzał się szukając dziewczyny. Nie było ani jej ani napastników. Po całonocnym poszukiwaniu zmęczony dowlókt się do domu. Rozumiał już, co było celem napaści. Nie widział szans odzyskania ukochanej. Pograżony w depresji myślał tylko o niej.

Pewnego dnia otrzymał list od pana X, z którego dowiedział się, że Sylvia porwana została przez starego i potężnego Ninję, i że jest więziona w ogromnym pałacu Ninji na najwyższym piętrze. Thomas natychmiast ruszył na pomoc ukochanej.

Pałac Ninji ukryty był wśród drzew. Thomas ostrożnie zbliżał się do wejścia. W pobliżu nie było żywej duszy. Mistrz ostrożnie wszedł do środka. Ledwo przestąpił próg, rozległ się trzask i brama zatrasnęła się. W tej chwili uchylły się drzwi w końcu sali. Thomas podbiegł do nich i zagłębił się w czeluść ciemnego korytarza. Powoli szedł naprzód omijając ukryte kłapy-pułapki. Nagle zauważył sunącego prosto na niego mężczyznę. Nie miał wątpliwości co do jego zamiarów, uprzedził więc jego atak. Jak spod ziemi wyrósł przed nim następny napastnik lecz tego Thomas także powalił.

Podobnych niespodzianek spotkał jeszcze wiele. Po ziemi pełzały węże, zniecacka wybuchały kolorowe ognie, dym gryzł w oczy a z ukrytych w ścianach drzwi wylatywały roje wirujących noży. Nic nie zdołało powstrzymać Thomasa przed nieustannym parciem naprzód. Pokonał w ten sposób cztery piętra i wreszcie znalazł się przed komnatą potężnego Ninji. Bez wahania wtargnął do środka. Tu czekała go najcięższa próba, z której też wyszedł zwycięsko. Mnisi z klasztoru Shao-Lin mogli być z niego dumni. A z Ciebie?

**Firma:** Ocean Software  
**Komputer:** Commodore 64/128, Spectrum 48/+1+2, Amstrad/Schneider. (mp)

## KRÓL I KRÓLOWA GIER



**JOASIA MOLENDĄ** lat 12, zamieszkała w Krakowie, ul. Siewna 21a m 61, jest uczennicą Szkoły Podstawowej Nr 109. Jej hobby to pływanie i stąd chyba ulubioną grą Joasi jest SUBMISSION na ATARI 800 XL.



**JANUSZ JUSZCZYK** lat 14, zamieszkały w Kaliszu, ul. Grunwaldzka 21, chodzi do VIII klasy Szkoły Podstawowej. W domu ma Amstrada i ulubioną grę — SKOOL DAZE. W wolnych chwilach oprócz grania i programowania zajmuje się elektroniką.



komputery  
osobiste

D. Madej, K. Marasek, K. Kuryłowicz

KOMPUTERY  
OSOBISTE

Z Wydawnictwami Komunikacji i Łączności związali się na dłużej trzej autorzy „Przewodnika po ZX Spectrum” Dariusz Madej, Krzysztof Marasek i Krzysztof Kuryłowicz. Kolejne ich dzieło — „Komputery osobiste” to skarbnica wiedzy o mikrokomputerach. Książka ta może być znakomitym prezentem również dla kogoś, kto nie ma zamiaru spędzać połowy dnia przy klawiaturze. Znajdziemy w niej ogromną ilość wiadomości i odpowiedzi na wiele pytań, które intrygują nie tylko informatyków. Przyda się też rodzicom, których dziecko w każdej chwili może poprosić o wyjaśnienie, co to jest mikrokomputer. Zawiera obszernie komentarze poświęcone sprzętowi, językom programowania, programom użytkowym. Ułatwia zrozumienie, że komputer jest bardzo wyrafinowanym urządzeniem, którego zastosowanie nie kończy się tylko na grach.

Wydawnictwu i autorom należą się gratulacje, że zdążyli umieścić w książce informacje niemal z ostatniej chwili, o twardych dyskach, dyskach optycznych, drukarkach laserowych, nowoczesnych pakietach oprogramowania czy najnowszym sprzęcie.

Dużą pomocą dla czytelnika są bardzo ładne i czytelne ilustracje, które często więcej mówią niż kilkunastoniowy wywód. Znakomitym tego przykładem jest np. rozdział poświęcony sieciom lokalnym. (j.j.)

**D. Madej, K. Marasek, K. Kuryłowicz, „Komputery osobiste”, Wydawnictwo Komunikacji i Łączności, Warszawa 1987, wydanie I, nakład 150 tys. egz. Cena 500 zł.**

ZAMIESZCZANIE  
NA EKRANIE

**Jednym z bardziej skomplikowanych układów cyfrowych w mikrokomputerach CPC Amstrad jest sterownik monitora ekranowego, odpowiedzialny za kontrolę generowanego sygnału wizyjnego.**

CRTC 6845 jest programowalny za pośrednictwem 19 wewnętrznych rejestrów, dostępnych przez dwukierunkowy 8-bitowy port, którego tylko dwa adresy (wrócimy do tego nieco później) wystarczą do zmiany zawartości tych rejestrów. Rejestry 16 i 17 odpowiedzialne są za obsługę pióra świetlnego, lecz są niedostępne dla użytkownika, zaś rejestr 18 pośredniczy w programowaniu 18 pierwszych. Interesują nas więc rejestry od 0 do 15.

Rejestr	Funkcja	Zakres
0	Całkowita ilość znaków w wierszu plus synchronizacja	0-255
1	Ilość wyświetlanych znaków w wierszu	0-255
2	Synchronizacja pozioma (pozycja)	0-255
3	Długość synchr. poziomej	0-15
4	Całkowita ilość wierszy w pionie	0-127
5	Synchronizacja pionowa	0-31
6	Ilość wyświetlanych wierszy	0-127
7	Synchronizacja pionowa (pozycja)	0-127
8	Tryb wyświetlania	0-3
9	Ilość linii w wierszu -1	0-31
10	Linia w wierszu, od której kursor kończy przesłanie znaku	0-31
11	Linia w wierszu, na której kursor kończy przesłanie znaku	0-31
12	Adres pamięci ekranu (bajt starszy)	0-255
13	Adres pamięci ekranu (bajt młodszy)	0-255
14	Pozycja kursora (bajt starszy)	0-255
15	Pozycja kursora (bajt młodszy)	0-255

Standardowe wartości wynoszą przykładowo: w MODE 1 R0=63, R1=40, R2=46 i R3=142. Parametry ekranu w pionie: R4=38, R5=0, R6=25 i R7=30.

Do zapisania danych w odpowiednich rejestrach wystarczy znać dwa adresy wspomnianego wyżej PORTu: &BC00 — w celu wybrania rejestru i &BD00 dla wprowadzenia pożądanego parametru. Wpisanie wartości „1” do rejestru 8 wymaga więc wprowadzenia (w BASIC-u) następującej sekwencji: OUT &BC00,8: &BD00,1. Nie zachęcam jednak do daleko idących eksperymentów, gdyż bez dokładnej znajomości wszystkich parametrów technicznych CRTC 6845 i umiejętności programowania można doprowadzić do ZNISZCZENIA układu. Niniejszy opis ma tylko ułatwić zrozumienie potencjalnych możliwości sterownika monitora, natomiast bezpieczne zmiany parametrów zapewni program „PRSCROLL.BIN” utworzony po wpisaniu i uruchomieniu programu ładującego „PRSCROLL.BAS”. Program należy wpisywać po wcześniejszym uruchomieniu „Weryfikatora” z poprzedniego numeru „Bajtki”. Liczby w nawiasach kwadratowych są

sumą kontrolną generowaną przez „Weryfikator” i służą tylko do sprawdzania poprawności wpisywania linii programu. Należy również pamiętać o włożeniu dyskietki do stacji przed uruchomieniem programu ładującego. W wyniku załadowania powstałego w ten sposób programu (LOAD „PRSCROLL.BIN”, &A000) uzyskujemy dwa sposoby zmiany organizacji wyświetlania ekranu: CAL&A000, ile razy, kierunek — gdzie „ile razy” określa krotkość przesunięcia linii całego ekranu, natomiast „kierunek” = 1 dla przesunięcia w górę i = 0 dla przesunięcia w dół; CAL&A000, lewy, prawy, górna, dolna, ile razy, kierunek — gdzie „lewy, prawy, górna, dolna” są parametrami określającymi fragment ekranu (tak jak przy określaniu parametrów WINDOW) a „ile razy, kierunek” mają podobne funkcje jak w poprzednim przykładzie. Krótki program demonstracyjny „DEMO” spowoduje nieco zamieszania wyłącznie dla usprawiedliwienia tytułu publikacji. Za zamieszczenie w głowach odpowiada niżej podpisany...

Wojciech Ziółtek

```

10 PRSCROLL (loader PRSCROLL.B [3550]
IN)
20 MEMORY &9FFF [134]
30 FOR i=&A000 TO &A039 STEP 8 [1020]
40 FOR j=i TO i+7:READ a# [994]
50 c=VAL("&"+a#):POKE j,c:NEXT j [3185]
EXT
60 SAVE "Prscroll.bin",b,&A000,& [2607]
39
70 DATA fe,02,28,26,fe,06,c0,dd [1302]
80 DATA 5e,04,1d,dd,6e,06,2d,dd [930]
90 DATA 56,08,15,dd,66,0a,25,dd [1630]
100 DATA 46,02,c5,e5,d5,af,dd,46 [982]
110 DATA 00,cd,50,bc,d1,e1,c1,10 [805]
120 DATA f1,c9,dd,46,02,c5,af,dd [1492]
130 DATA 46,00,cd,4d,bc,c1,10,f5 [2026]
140 DATA c9,00,00,00,00,00,00,00 [1564]
150 END [110]

```

```

10 'Program demonstracyjny DEMO [2678]
20 LOAD "prscroll.bin",&A000 [2796]
30 MODE 1:CLS: [1104]
40 FOR b=1 TO 15:PRINT "CALL &A0 [4542]
00,ilerazy,kierunek":NEXT b
50 FOR i=1 TO 4 [450]
60 CALL &A000,i,0 [888]
70 FOR a=1 TO 100:NEXT a [1143]
80 CALL &A000,i,1 [891]
90 FOR a=1 TO 100:NEXT a [1143]
100 NEXT i [375]
110 MODE 2:FOR c=1 TO 15 [1950]
120 PRINT "CALL &A000,lewy,prawy [4931]
,gorna,dolna,ilerazy,kierunek"
130 NEXT c [381]
140 FOR i=10 TO 40 STEP 10 [699]
150 CALL &A000,1,i,2,i,3,0 [1359]
160 FOR a=1 TO 50:NEXT a [1208]
170 CALL &A000,1,i,2,i,3,1 [1358]
180 FOR a=1 TO 50:NEXT a [1208]
190 NEXT i [375]

```

## NAPĘD 5 1/4" DO CPC 6128

**Komputery Amstrada CPC 6128 sprzedawane są z jedną 3 calową stacją dysków. Producent przewidział możliwość dołączenia drugiego takiego samego napędu. Z pewnych względów bardziej użyteczne może się okazać zastosowanie stacji 5 i 1/4 cala.**

Do przeróbki potrzebne są dwa 34-stykowe złącza krawędziowe, pół metra 34-żyłowego kabla, dwa przełączniki i dodatkowy zasilacz dwunapięciowy: 12V 700mA, 5V 500mA. Oczywiście niezbędny jest też typowy 5 calowy napęd dysków np. TEAC 360 KB.

Konieczne połączenie między stacją a komputerem przedstawiono na rys.1. Przełączniki „READY„ i „HEAD” zapewniają właściwą pracę urządzeń. Sygnał READY podawany ręcznie służy do poinformowania komputera, że w stacji znajduje się dyskietka. Poza programami typu DISCKIT przełącznik ten powinien być stale zwarty. Sygnał „HEAD”, także podawany ręcznie, pozwala na dostęp do obu stron dyskietki bez konieczności jej przekładania i robienia dodatkowych dziurek w kopercie.

Jonasz Mayer



Rys. 1 — Połączenie między stacją dysków a komputerem Amstrad CPC 6128

1. +12 V      700 mA
2. gnd
3. gnd
4. +5 V      500 mA

Rys. 2 — Łącze zasilające stacji dysków

# AMSTRAD INFO

**Jeśli zamierzamy pisać programy, mogące różnie działać w zależności od typu komputera, na którym są uruchamiane, to przedstawiony poniżej program ułatwi nam zrobienie tego. Jeśli nie, to i tak problemy poruszone w tym krótkim artykule mogą okazać się interesujące dla posiadaczy 8-bitowych komputerów firmy Amstrad.**

Prezentowany program składa się z dwóch części: jednej napisanej w Turbo-Pascalu i drugiej w assemblerze. Program pascalcowy zawiera dwie procedury VERSION i INFO. Każda z nich wypisuje dane o komputerze po wcześniejszym odwołaniu się do systemu operacyjnego poprzez funkcję RSX'a, dostępne dzięki krótkiemu programowi w assemblerze.

W procedurze VERSION rekord INFO ma następujące znaczenie:

- a=0 — komputer: CPC 6128
- a<>0 — komputer: PCW 8256/8512
- b,c — zawierają numer wersji BIOS'a
- h,l — zawierają numer wersji ROM'u (istotne tylko dla CPC, ponieważ PCW nie ma żadnego ROM'u).

W procedurze INFORMATION rekord ten ma następujące znaczenie:

- a=0 — 1 stacja dysków
- a<>0 — 2 stacje,
- b — liczba bloków pamięci, każdy ma 16 KB
- c=0 — brak interfejsów, w przypadku CPC oznacza brak tylko RS-232C
- c<0 — oznacza połączony RS-232C dla CPC, a dla PCW oznacza także podłączenie złącza CENTRONICS.

Narzędzia potrzebne do pracy są następujące:

1. Turbo-Pascal — kompilator i edytor,
  2. M80 — assembler (jeśli nie mamy, to warto się w niego zaopatrzyć, bo dostępny na dyskietce firmowej RMAC nie akceptuje kodów Z80),
  3. GENCOM.COM — umożliwia tworzenie RSX-ów i ich łączenie ze zbiorami typu .COM
  4. LINK.COM — konsolidator (ang. LINKER)
- Zbiory GENCOM.COM i LINK.COM znajdują się na dyskietkach systemowych.

Metoda postępowania:

1. przy pomocy edytora Turbo-Pascal tworzymy dwa zbiory tekstowe:
  - a. INF.PAS — wersja źródłowa programu pascalcowego,
  - b. INF.MAC — wersja źródłowa programu assemblerowego.
2. kompilujemy na dysk zbiór INF.PAS wybierając adres końcowy C000 H przy pomocy opcji 0 i E (patrz opis Turbo-Pascal!), otrzymujemy zbiór INF.COM
3. na zbiór INF.MAC działamy assemblerem M80:
 

```
A>m80=inf <RETURN>
```
4. wykonujemy operacje:
 

```
A>link inf op
A>ren inf.rsx=inf.prl
A>gencom inf inf
```

W wyniku poprawnego wykonania tych operacji uzyskujemy zbiór INF.COM, który po uruchomieniu na komputerze da na ekranie wydruk podobny do przedstawionego niżej.

Gotowy program INF nie ma oczywiście większego znaczenia poza prostą demonstracją, ale jeśli dołączymy procedury pascalcowe do własnego programu napisanego w tym języku i wykonamy podobne operacje do opisanych wyżej, to uzyskamy własny program, który może korzystać z istotnych informacji systemowych. Jedyną zmianą jest zastosowanie operacji:

A>gencom WLASNY inf  
jeśli nasz program pascalcowy nazywa się WLASNY.PAS i po skompilowaniu dostajemy zbiór WLASNY.COM.

Opisana metoda może być przeniesiona, także na programy w języku BASIC, ale jest to bardziej kłopotliwe, co jest kolejnym argumentem w ogólnej dyskusji, jakiego używać języka programowania.

Przedstawiony zestaw narzędzi i metoda postępowania stanowi idealne rozwiązanie przy tworzeniu własnych RSX-ów.

```
Informacja o komputerze
AMSTRAD PCW 8256/8512      BIOS version: 1,4
Konfiguracja
1. Pamięć RAM 512 KB
2. 1 stacja dysków
3. brak interfejsów
```

Przykładowy wydruk programu:

Jonasz Mayer

## LISTING 1

```
program info.version;
(*****
(* program dla komputerów 8-bitowych firmy Amstrad *)
(* pozwalający określić komputer i jego konfigurację *)
(* (C) J. M. grudzień 1987 *)
(* uwaga: korzysta z funkcji RSX'a o numerze 69 *)
(*****
const
  cdvers = $00E3;
  cdinfo = $00E6;
  rsxnr = 69; (* 45H *)
type
  HexStr = string(4.);
  info = record
```

```

  a : byte;
  b,c : byte;
  h,l : byte;
end;
var
  pinfo : ^info; PCW : boolean;
procedure Version;
begin
  pinfo := ptr (BdosHl(rsxnr,cdvers));
  with pinfo^
  do begin
    writeln; PCW := false;
    if a=0
    then write(' AMSTRAD CPC 6128 ')
    else begin
      write(' AMSTRAD PCW 8256/8512 ');
      PCW := true;
    end;
    write(' BIOS version: ',b:l,',',c:l);
    if not PCW
    then write(' ROM version: ',h:l,',',l:l);
  end;
  writeln;
end; (* version *)
procedure information;
begin
  pinfo := ptr (BdosHL(rsxnr,cinfo));
  with pinfo^
  do begin
    writeln;
    writeln(' Konfiguracja:');
    writeln(' 1. Pamięć RAM ',b:l*4,' KB');
    if a=0
    then writeln(' 2. 1 stacja dysków')
    else writeln(' 2. 2 stacje dysków');
    if c=0
    then if PCW
      then writeln(' 3. brak interfejsów')
      else writeln(' 3. brak RS-232C')
    else if PCW
      then writeln(' 3. Centronics i RS-232C')
      else writeln(' 3. RS-232C');
  end;
end; (* of information *)
begin
  writeln(' Informacja o komputerze ':40);
  version;
  information;
end.

zbiór INF,PAS
```

## LISTING 2

```

*****
* rsx dostępu do informacji o sprzecie *
* (C) J. M. grudzień 1987 *
* dane wejściowe: *
* rejestr DE zawiera numer procedury systemowej *
* 1, E6 - cd_info *
* 2, E3 - cd version *
*****
        ,z80
rsxnr equ 69 ; 45H
cdinfo equ 00E6H
cdvers equ 00E3H
warm equ 0001H
userf equ 3*(30-1)
;
; cseg
; rsx header
ds 6
jp start
next:jp 0000h
dw 0
db 0FFH
db 0
db 'infovers'
db 0,0,0
start:
push af
ld a,c
cp rsxnr
jp z,begin
pop af
jp next
begin:
ld (func),de
ld hl,(warm)
ld bc,userf
add hl,bc
call jphl
func:dw 00E6H ; E6 info, E3 vers
ex de,hl
ld hl,buffer
ld (hl),a
inc hl
ld (hl),b
inc hl
ld (hl),c
inc hl
ld (hl),e
inc hl
ld (hl),d
ld hl,buffer
pop af
ret
jphl:jp (hl)
;
buffer:
ds 5
end
; zawartość zbioru INF,MAC
```

# NASTĘPNY KROK

**System operacyjny czyli program nadzorujący pracę komputera. Już dwa miesiące śledziliśmy jego rozwój i dziś kontynuujemy to zajęcie. Pamiętamy, że rozwój oprogramowania systemowego był wymuszany przez lenistwo obsługi oraz buźliwy rozwój możliwości sprzętu.**

Pożegnaliśmy się w sytuacji dość niejasnej. Procesor miał czytać dane nie bezpośrednio z czytnika kart, lecz za pośrednictwem znacznie szybszych pamięci taśmowych czy dyskowych<sup>\*)</sup>. Kto w takim razie ma je wczytać do tych pamięci? Jedno z możliwych rozwiązań było następujące: małe, tani komputerki wczytują paczkę kart perforowanych, stanowiących kompletne zadanie<sup>\*\*</sup>) na taśmie magnetyczną. Po niej następną i następną, tworząc w ten sposób na taśmie tzw. wsad zadań, który następnie przenoszono na duży komputer, ten zaś zadania po kolei realizował, a wyniki w podobny sposób (za pośrednictwem taśmy) wracały do wydrukowania na drukarce. System operacyjny dużego komputera nadzorował przejście od jednego zadania do drugiego.

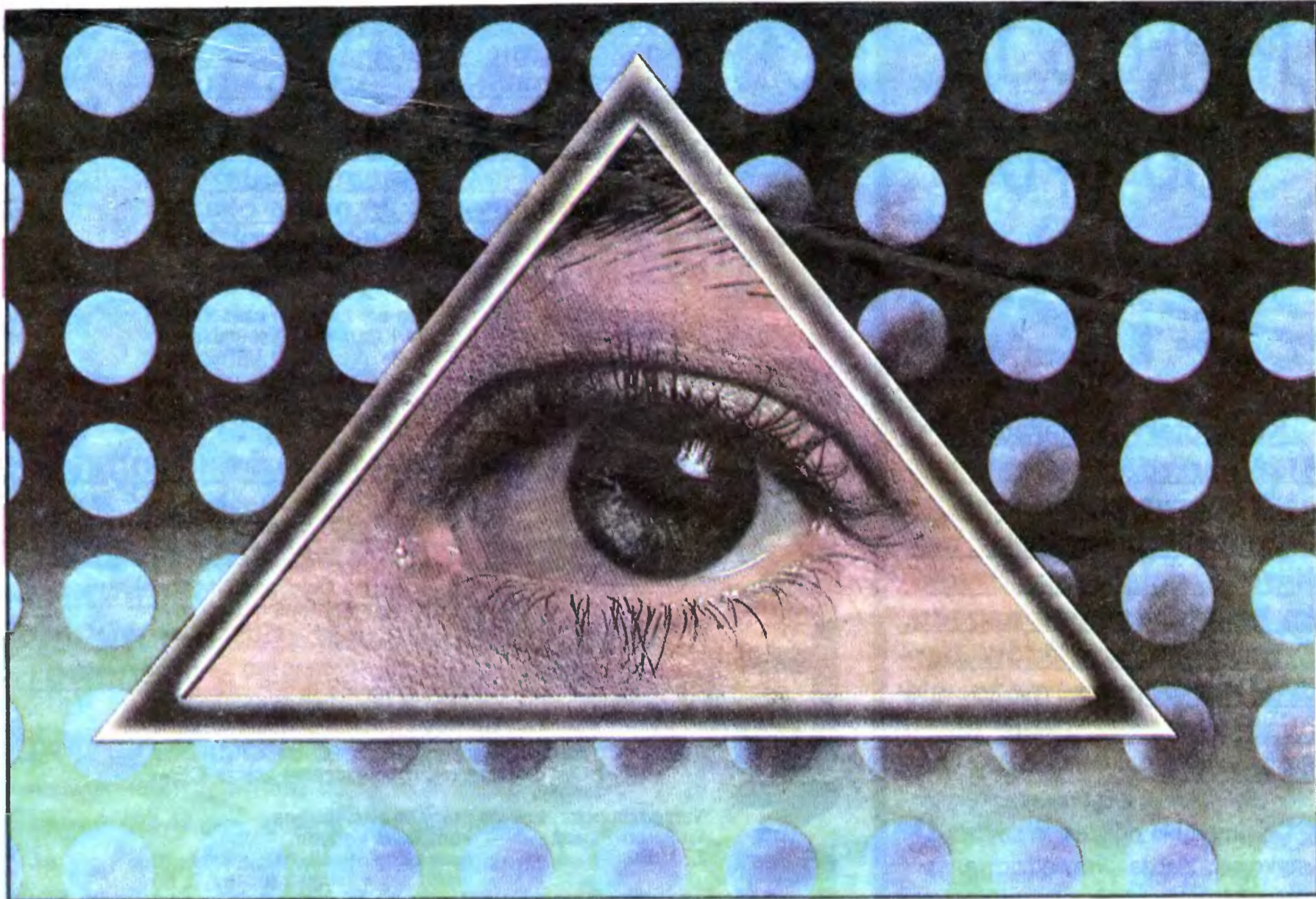
Rozwiązanie to bardzo zwiększało wykorzystanie drogiego, szybkiego procesora oraz urządzeń wejścia/wyjścia, kosztem niewielkiej rozbudowy sprzętu. I tu zaskakujący zwrot sytuacji: drugim składnikiem kosztu była ogromna niedogodność tego systemu przetwarzania dla człowieka — separacja od komputera, długie oczekiwanie na wyniki, skomplikowane zabiegi z noszeniem mnóstwa taśm. Bardzo szybko okazało się, że takiego kosztu nikt nie chce ponosić. Musiało więc powstać lepsze rozwiązanie, ale o nim za chwilę, gdyż będzie nam łatwiej je zrozumieć gdy przejdziemy przez następny etap rozwoju. Czytelnik, który zechciał poświęcić odrobinę uwagi uprzednim odcinkom, z łatwością zgadnie, że procesory komputerów znów przyspieszyły.

## SZYBKI PROCESOR — WOLNY DYSK

W związku z tym dyski stały się stosunkowo wolne i w oczekiwaniu na transmisję danych procesor wylegiwał się bezczynnie np. przez całą jedną dziesiątą każdej sekundy (jeżeli się robi kilka milionów operacji na sekundę, to jest to naprawdę poważna przerwa).

Rozwiązanie jest następujące: trzeba umieścić w pamięci operacyjnej komputera kilka programów na raz (oczywiście w oddzielnych obszarach). Jeśli jeden z nich zażąda przestania danych z dysku, to procesor nie czeka aż to nastąpi, tylko zaczyna wykonywać inny program, wybrany spośród czekających w pamięci. Genialnie proste, wejźmy jednak w skórę tego, który ma tę wieloprogramowość zrealizować w praktyce.

Problemów jest mnóstwo: skoro w momencie żądania transmisji procesor ma zacząć liczyć inny program, to potrzebny jest ktoś, kto tę transmisję zrealizuje i umieści dane z dysku w pamięci operacyjnej. Czyli pojawiają się nowe elementy architektury komputera: wyspecjalizowane urządzenia do sterowania wejściem/wyjściem, noszące nazwę kanałów; dalej bloki DMA (Direct Memory Access — bezpośredniego dostępu do pamięci), pozwalające kanałom pisać i czytać pamięć bez udziału procesora. Wreszcie, gdy transmisja już została zakończona trzeba procesor zawiadomić — a więc rozbudowany system impulsów przerywających normalny tok pracy,



# SYSTEM OPERACYJNY

## Cz. III

czyli po prostu przerwać).

No i jeszcze „drobiazg” — jeśli w ciągu jednej sekundy procesor przełącza się na obsługę kilkunastu różnych programów, to na pewno przełączaniem tym nie steruje człowiek! A kto? SYSTEM OPERACYJNY. Napisałem go drukowanymi literami, bo teraz to naprawdę ma co robić.

Oczywiście małe komputerki przepisujące karty perforowane na taśmę znikają automatycznie. Ich funkcje wykonują kanały. Tak więc komputer ma z powrotem swoje urządzenia wejścia/wyjścia, ale podłączone nie bezpośrednio do procesora, ich pracą sterują kanały. Skoro umiemy już godzić wczytywanie danych z wolnych czytników z efektywnym wykorzystaniem procesora, to możemy pójść jeszcze dalej i dołączyć do naszego systemu cyfrowego<sup>\*\*\*</sup>) końcówki, składające się z klawiatury i monitora, pozwalające człowiekowi kontaktować się z maszyną bez pośrednictwa kart czy taśm perforowanych. Dzięki swej ogromnej szybkości jeden procesor może obsługiwać równocześnie wielu użytkowników (wiele końcówek) i każdemu z nich wydaje się, że ma cały komputer dla siebie. Taki sposób eksploatacji nazwano wielodostępem. Typową praktyką jest również instalowanie końcówek zdalnych — nie stojących bezpośrednio przy komputerze, lecz np. w innej miejscowości i współpracujących za pośrednictwem łącz telekomunikacyjnych.

## DUŻE JEST SKOMPLIKOWANE

Wszystkie elementy muszą współdziałać w efektywny i harmonijny sposób, co

nakłada na system operacyjny ogromne zadania. Musi on rejestrować wszystkie wpływające zadania, ustalać kolejność w jakiej będą wykonywane, jeśli zadanie wczytane z kart nie może być wykonane od razu to przechować je na dysku, rozsyłać wyniki na odpowiednie drukarki, zarządzać pamięcią składającą się z wielu stacji dysków i taśm, a wszystko to z absolutną precyzją — przekłamanie jednego adresu transmisji może dezorganizować cały system. Trzeba też dopilnować, aby żadne zadanie nie czekało zbyt długo, udostępniać wszystkim użytkownikom programy publiczne (np. edytory, translatory, bazy danych itd.), wreszcie za wszystkie wykonane operacje obliczać i wystawiać rachunki użytkownikom.

Dodatkowo dochodzą problemy ochrony użytkowników przed sobą (w tej samej pamięci siedzą obok siebie programy różnych ludzi a każdy ma swoje tajemnice), oraz systemu przed użytkownikami. Łatwo sobie wyobrazić pełną dezorganizację systemu, spowodowaną przez użytkownika, który dla eksperymentu zastąpił jeden z systemowych podprogramów obsługi zbiorów, własnym podprogramem, i (np. przez pomyłkę) w miejscu, w którym była zapisana kartoteka zbiorów, zapisał list do narzeczonych.

Aby taka ochrona była skuteczna potrzebne są nowe mechanizmy sprzętowe, np. pamięć operacyjna, która automatycznie blokuje próbę zapisu lub odczytu poza przydzielonym dla jednego programu obszarem.

## CO MOŻE MIKRO?

Oczywiście oprócz ogromnych systemów cyfrowych budowano (i buduje się) także mniejsze, w których część z opisanych elementów w ogóle nie występuje. Jednak ich konfiguracje są ciągle o wiele bardziej rozbudowane niż przeciętny mikrokomputer, szczególnie domowy.

Mikrokomputery nie powstały jako zmminiaturyzowane wersje dużych systemów cyfrowych<sup>\*\*\*\*</sup>), nie odziedziczyły ani ich pełnej architektury ani systemów operacyjnych. Początkowo w ogóle nie miały żadnego systemu operacyjnego, także programowanie odbywało się w języku maszynowym...

Jeśli ktoś zrobił mi ten zaszczyt i przeczytał uważnie dwa poprzednie odcinki, to pamięta, że tak właśnie zaczynała się historia systemów operacyjnych dużych komputerów. Historia, którą opowiadałem Wam nie przypadkiem, gdyż ona teraz, na naszych oczach, powtarza się dla sprzętu mikro!

Na prostych, zwykle nie posiadających pamięci dyskowych, maszynkach domowych polecenia sterujące pracą maszyny wmontowane są w BASIC (komendy RUN, SAVE itp.).

Ogromną popularność zdobył system CP/M, który nie tylko jest nadzorcą maszyny, lecz organizuje na dyskietce własny, prosty system plików, oraz udostępnia programom użytkownikom zestaw systemowych procedur, obejmujących wejście/wyjście.

Wraz z rozpowszechnieniem się sprzętu mikro i rozwojem jego możliwości wzrastają apetyty użytkowników i

# YRUFU 8B UFORY

oczywiście rozwijają się również systemy operacyjne. Mogą one obsługiwać równocześnie kilka urządzeń. Np. w systemie DOS \*\*\*\*\*) dla IBM PC można zlecić wydruk zawartości zbioru dyskowego i zająć się innym zadaniem — drukowanie odbywa się równoległe z przetwarzaniem następnego zadania.

Równocześnie pojawiają się liczne próby realizacji w oparciu o mikrokomputery systemów wieloprogramowych i wielodostępnych. Nie zawsze są one udane — moc obliczeniowa procesora jest czasami niewystarczająca, brakuje odpowiednio rozbudowanego systemu przerwań i autonomicznego wejścia/wyjścia, dynamicznego zarządzania pamięcią operacyjną. Jednak sam fakt podejmowania takich prób (oczywiście w połączeniu z ciągłym postępem technologii — „komputery przyspieszają”) skłania producentów do rozbudowy konfiguracji nowych modeli o elementy ułatwiające np. wieloprogramowość.

Istnieją już systemy wielodostępne dla mikrokomputerów 16-bitowych, najbardziej znane z nich to XENIX i UNIX, będące przeniesieniem bardzo popularnego na minikomputerach systemu UNIX. Mogą one działać np. na IBM PC AT i zachynają zdobywać popularność także w Polsce.

Czego możemy spodziewać się w niedalekiej przyszłości, czy rozwój sprzętu i oprogramowania w skali mikro jest i będzie wierną kopią rozwoju dużych maszyn? Myślę, że nie. Historia powtarza się, ale nigdy dosłownie. Przecież konstruktorzy i informatycy znają poprzednie rozwiązania i wiedzą, które dobrze sprawdzały się w praktyce, a które warto wyeliminować. Z drugiej strony ciągły rozwój elektroniki może w każdej chwili otworzyć nowe, dotychczas niezbrane możliwości.

Inny aspekt to odbiorcy sprzętu. Przednio komputery były programowane i obsługiwane przez stosunkowo nieliczną grupę fachowców — przeciętny klient stawiał zadanie i otrzymywał wyniki. Teraz nastąpiło znaczne umasowienie, każdy sam obsługuje swój komputer osobisty. Stawia to nowe wymagania przed oprogramowaniem. Musi ono być łatwe w obsłudze nawet dla laika, stąd burzliwy rozwój nowych koncepcji komunikacji operatora z programem, wymyślne menu, ikony itd.

Generalnie jednak droga do osiągnięcia przez mikro pełnych możliwości dużych instalacji jest jeszcze daleka, dlatego warto zachować trochę dystansu do twierdzenia: „komputery wkroczyły w nasze życie bo pojawiło się Spectrum”. Mnie osobiście inwazja sprzętu mikro uświadamia gdzie nie mamy komputerów, chociaż powinniśmy mieć i to już dawno!

Na koniec wreszcie chcę wyrazić nadzieję, iż przekonał się wspólnie o tym jak wiele zadań wykonuje za nas system operacyjny, a co za tym idzie o jego wartości i niezbędności, a także o tym, że zrobienie dobrego systemu operacyjnego to duża sztuka.

\*) Mamy tu przykład buforowania: szybka pamięć magnetyczna stanowi bufor między wolnym czytelnikiem a bardzo szybkim procesorem.

\*\*\*) Treść programu, dane itd.

\*\*\*\*) Przy tak rozbudowanych instalacjach bardzo często zamiast komputer mówimy właśnie system cyfrowy.

\*\*\*\*\*) Ich przodków należy szukać wśród elektronicznych kalkulatorów.

\*\*\*\*\*) Uwaga na nazwę DOS. Jest to angielski skrót terminu Disk Operating System (dyskowy system operacyjny) i wielu autorów używa go jako nazwy swojego systemu. Stąd DOS można spotkać na wielu różnych typach komputerów, oczywiście oznacza to za każdym razem zupełnie inny system.

Andrzej Pilaszek

## Zanim zajmiemy się informatyką przypomnijmy sobie dokładne znaczenie słowa bufor w zastosowaniach tradycyjnych. Z grubsza rzecz biorąc jest to urządzenie łagodzące wstrząsy przy zetknięciu dwóch pojazdów poruszających się z różnymi prędkościami. Zaraz zobaczymy czy przechodząc do informatyki słowo to zachowało podstawowe znaczenie.

Przyjrzyjmy się współpracy procesora z klawiaturą. Procesor wykonuje rozkaz „Czytaj znak z klawiatury” \*). Jeśli jest wciśnięty jakiś klawisz, to odpowiedni kod zostanie wczytany, czyli umieszczony w umówionym miejscu pamięci komputera. Teraz drobna dygresja: kiepski program po wykonaniu rozkazu czytania idzie dalej i po prostu przetwarza pobrany (?) znak. Dobry sprawdzi, czy rzeczywiście coś zostało wczytane, bo może żaden klawisz nie był wciśnięty. W tym wypadku zaczyna działać w pętli — wykonuje wczytanie i sprawdzenie poprawności aż do momentu faktycznego wczytania znaku. Unika w ten sposób przetwarzania śmieci. Takie sprawdzanie poprawności wykonania operacji jest niezbędne gdy sami programujemy obsługę urządzenia (np. w języku wewnętrznym maszyny).

Jeśli natomiast komunikujemy się z nim za pośrednictwem procedur wejścia/wyjścia dostarczonych przez język programowania czy system operacyjny, to problemy te są już załatwione. Np. instrukcja INPUT w BASIC-u czeka aż operator wpisze dane — nie musimy w naszym programie umieszczać pętli oczekiwania na pojedyncze znaki.

Wróćmy do wczytywania: klawisz wciśnięty, znak wczytany. Jeśli jednak, po wczytaniu każdego znaku program musi wykonać dłuższe rachunki, to może się okazać, że operator pisze szybciej niż komputer czyta i zanim jeden znak zostanie wczytany naciśnie już następny. Oczywiście część znaków ginie. Chyba że wyposażymy klawiaturę w niewielką pamięć, w której będą zapisywane naciskane klawisze. Procesor komputera będzie pobierał znaki nie bezpośrednio z klawiatury, lecz z tej pamięci, czyli właśnie BUFORA. Pamięć buforowa służy więc do amortyzacji różnicy prędkości między pisaniem a wczytywaniem znaków (w omówionym przypadku różnica na korzyść operatora). W przypadku klawiatury typowe długości bufora to 8 lub 16 znaków.

## KONIEC GUBIENIA ZNAKÓW

Oczywiście w dalszym ciągu istnieje możliwość przepelnienia bufora, np. jeśli program liczy przez godzinę i nic nie wczytuje. Dlatego niezbędny jest układ sygnalizujący brak miejsca na nowe znaki i blokujący wprowadzanie. (Zwykle w takiej sytuacji komputer reaguje piskiem na każde naciśnięcie klawisza).

UWAGA: nie należy mylić zawartości bufora klawiatury, ze znakami, które już są napisane i odbite na ekranie, lecz jeszcze nie zostały przesłane do przetwarzania naciśnięciem ENTER (lub RETURN). Znaki siedzące w buforze klawiatury nie zostały jeszcze zauważone przez resztę komputera, „wyskoczą na ekran” gdy

procesor przestanie być zajęty i poświęci im uwagę.

Wniosek praktyczny jest następujący: jeśli pracujemy z programem konwersacyjnym, który sterujemy np. naciskając spację, i program, obliczając coś, przestał reagować na klawiaturę, to nie warto wciskać tej spacji wielokrotnie. Pierwsza już siedzi w buforze klawiatury i czeka na pobranie. Następne też tam trafiają i będą po kolei pobierane i interpretowane jako następne polecenia, co zwykle wymiennie wyprowadza program w maliny.

## WĄSKIE GARDŁO — WOLNA DRUKARKA

Popatrzmy teraz co się dzieje na wyjściu, czyli na współpracę komputer — drukarka. Komputer jest w stanie produkować wyniki dużo szybciej niż drukarka przenosi je na papier. Często jednak produkcja tych wyników jest nierównomiernie rozłożona w czasie. Rozpatrzmy następujący przykład:

pewien program rozwiązuje 10 równań. Obliczenia dla jednego równania trwają 100 sekund, a po ich zakończeniu wysyła na drukarkę 10 linii wydruku (czas przestania komputer — drukarka będziemy systematycznie pomijać). Wydruk jednej linii zajmuje drukarce 5 sekund.

Po otrzymaniu znaku drukarka wysyła komputerowi sygnał: „Brak gotowości do przyjmowania znaków”, czym zmusza go do oczekiwania. Następny znak można wysłać dopiero gdy drukarka napisze poprzednio wysłany i wyśle komputerowi sygnał: „Gotowe, wysyłaj”. Przy tym schemacie obliczenia i wydruk dla jednego zestawu zajmują:

$$100s + 10linii * 5s = 150s$$

$$\text{a dla dziesięciu zestawów 1500s.}$$

Jeśli jednak możemy wysyłać na raz całą linię (drukarka ma bufor na tyle znaków ile mieści się w linii), a następną linię gdy tylko skończy się druk poprzedniej, to komputer musi czekać na drukarkę tylko do momentu wysłania ostatniej linii, czyli podczas drukowania dziesięciu, razem  $9 * 5s = 45s$ , a czas wykonywania obliczeń wynosi  $10 * (100s + 45s) = 1450s$ , a więc nieco lepiej.

Zwróćmy uwagę, że bufor został wprowadzony jakby między urządzenie bezpośrednio drukujące a komputer. Komputer wpisuje swoje dane do bufora z jednej strony, a z drugiej strony tegoż bufora są one pobierane do druku.

Co się stanie, jeśli powiększymy pojemność bufora drukarki do pełnych dziesięciu linii? Spróbujcie się najpierw zastanowić sami. Gotowe? To jedziemy. Po zakończeniu obliczeń dla jednego zestawu, komputer zaczyna przysyłać wyniki na drukarkę. Dopóki bufor nie będzie pełen drukarka przyjmuje je bez problemów, czyli (w bardzo krótkim czasie, któ-

ry pomijamy) wszystkie dane zostały wprowadzone. Przynajmniej z punktu widzenia procesora, który natychmiast przystępuje do następnych rachunków. W tym czasie drukarka, pracując równoległe z procesorem, przenosi zawartość bufora na papier. Czas wykonywania obliczeń:  $10 * 100s = 1000s$ . Po tym czasie komputer może rozpocząć następnego zadania. Na koniec drukowania musimy poczekać jeszcze dodatkowe 50s.

Efekt ostateczny: poprawa efektywności wykorzystania komputera o prawie jedną trzecią, a więc bardzo znaczna.

Koszty: wyposażenie drukarki w dodatkową pamięć, wielkości rzędu kilku kilobajtów oraz układy obsługi bufora. Przy dzisiejszych cenach pamięci i elektroniki są to koszty nieznaczące.

Myślę, że już nikt nie ma wątpliwości: jak to jest możliwe? Program już się skończył, widać to na monitorze, a drukarka jeszcze pracuje.

## TROCHĘ DODATKOWYCH KŁOPOTÓW

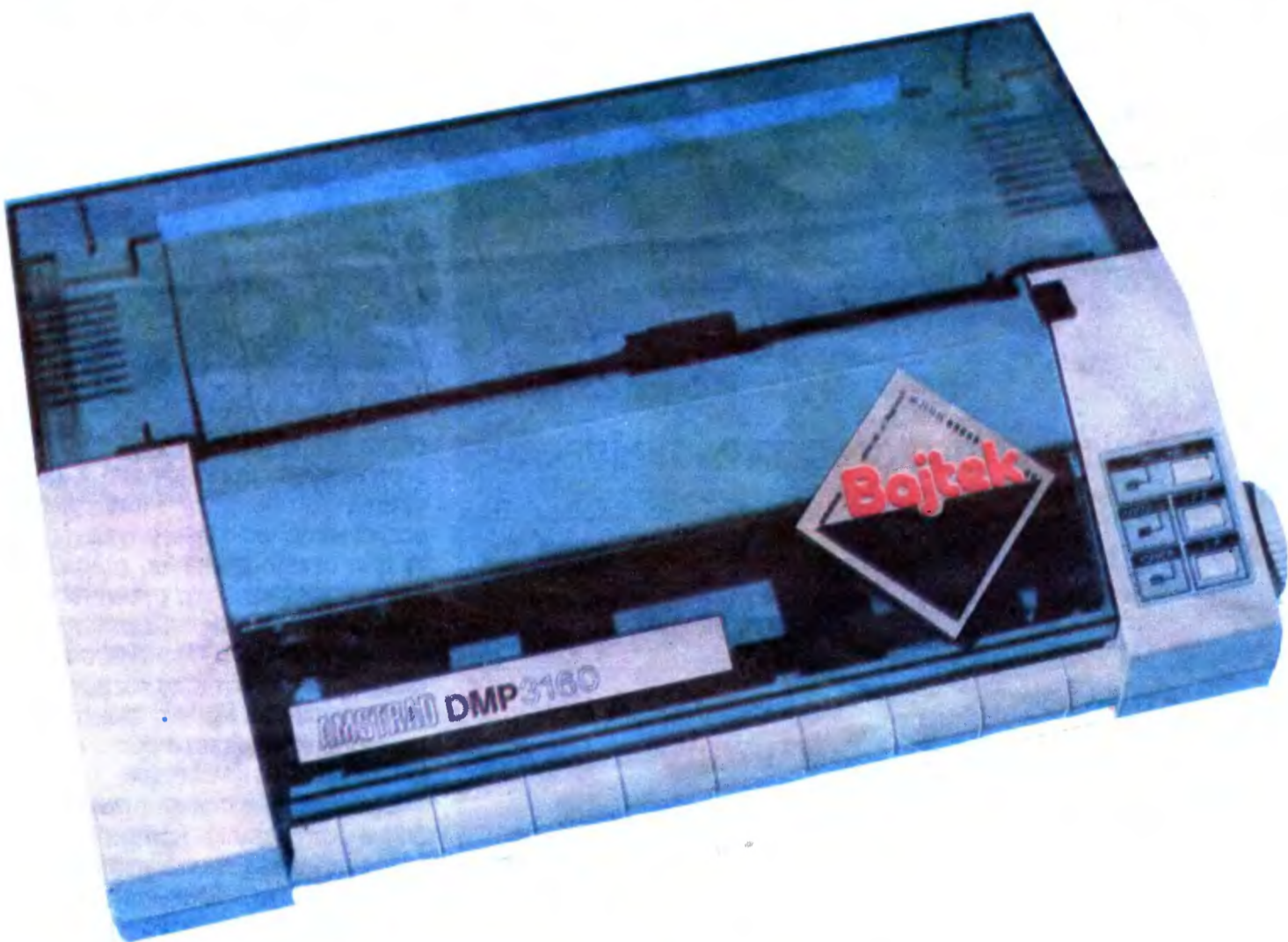
Rola bufora drukarki jest jednak bardziej złożona. Bywa bowiem całkiem odwrotnie — komputer generuje po kilka znaków, w sporych odstępach czasu. Mogą one być gromadzone w buforze drukarki i wyprowadzane na papier dopiero gdy linia zostanie zapełniona, lub komputer prześle bezwzględny rozkaz: „Koniec linii”. Właściwie nie ma to dla nas znaczenia, ale czasami zdarza się tak, że program wysła część znaków na drukarkę, lecz zanim zapełni lub zakończy linię, jego realizacja zostaje przerwana, np. przez wystąpienie błędu. Wtedy część znaków „znika”, wydaje się, że działanie programu zostało przerwane nie tam, gdzie faktycznie to nastąpiło i szukamy błędu nie tam gdzie jest naprawdę. Piszę o znikaniu w cudzysłowie, gdyż tak naprawdę, to te znaki są. Gdzie? Oczywiście w buforze drukarki. Naprawdę mogą zniknąć dopiero gdy wyłączymy ją z sieci. Jeśli nie, to brakujące znaki pojawią się na początku następnego wydruku.

Podobna sytuacja może wystąpić jeśli po ostatejniej instrukcji pisania w programie nie wyprowadzamy znaku końca linii. Może ale nie musi — dobry kompilator lub interpreter po napotkaniu instrukcji kończącej program, może spowodować opróżnianie buforów wszystkich urządzeń.

Buforowanie nie ogranicza się tylko do drukarki i klawiatury. Wprost przeciwnie, jest to technika bardzo rozpowszechniona. Stosujemy ją wszędzie tam, gdzie współpracują ze sobą urządzenia o różnych prędkościach działania, aby zoptymalizować wykorzystanie urządzeń, zmniejszyć czas wykonywania obliczeń i wreszcie zapobiec gubieniu informacji, gdy urządzenie wolniejsze nie nadąży i nie jest w stanie zatrzymać pracy szybszego.

\*) Oczywiście jest to umowne uproszczenie, zwykle takiego rozkazu nie ma, a przeczytanie znaku wymaga wykonania ciągu kilku rozkazów procesora.

Andrzej Pilaszek



Drukarke Amstrad DMP3160 otrzymaliśmy do testowania dzięki uprzejmości pana Andrzeja Łukomskiego, właściciela firmy wysyłkowej Polanglia będącej jedynym oficjalnym przedstawicielem Amstrada w Polsce.

## DRUKARKA AMSTRAD DMP 3160

**DMP3160 jest 9-igłową drukarką mozaikową. Jest ona przeznaczona zasadniczo do pracy z komputerami Amstrad PC oraz innymi komputerami kompatybilnymi z IBM PC. Może także współpracować z Amstradami serii CPC, a przy użyciu interface'u Centronics również z innymi komputerami domowymi (Atari XL/XE, Commodore 64/128, Spectrum 48/+/+2/+3).**

### BUDOWA I DZIAŁANIE

Już na pierwszy rzut oka odróżnia się od innych drukarek nietypową konstrukcją. Papier jest podawany od przedniej strony i przesuwany poziomo bez opasywania wałka, a więc odwrotnie niż w najczęściej dotychczas spotykanych modelach. Umożliwia to drukowanie nawet na dość grubym papierze o niewielkich wymiarach (np. koperty pocztowe), co normalnie jest raczej kłopotliwe. Przy korzystaniu z papieru perforowanego w bloku lub z papieru w rolce można drukarkę unieść do góry na specjalnych, składanych nóżkach i papier umieścić pod nią. Wałek o standardowej długości jest wyposażony w prowadnicę papieru i umożliwia użycie papieru o szerokości od 100 do 255 mm. Oczywiście dopuszczalne są wszystkie formy papieru: perforowany, z rolki lub w arkuszach. Jednorazowo można drukować oryginał z dwiema kopiami.

Sposób prowadzenia papieru wymusił pionowe ustawienie głowicy drukującej. Jest ona przesuwana po prowadnicy przez silnik krokowy za pośrednictwem paska zębatego. Taśma jest umieszczona w bardzo nietypowej, dwuczęściowej kasecie w taki sposób, że przebiega nad i pod głowicą

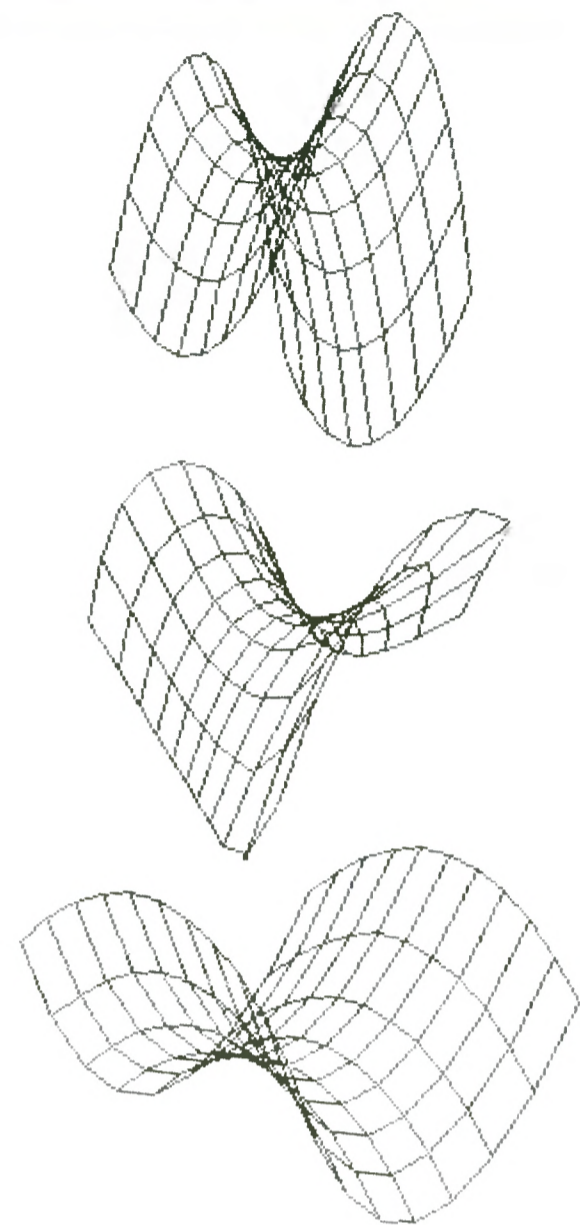
całą drukującą i jest naciągana sprężyną umieszczoną w jednej z połówek kasety. Przesuw taśmy wymusza jednokierunkowy zaczep na obudowie głowicy. Samo drukowanie odbywa się poza wałkiem na płaskiej podkładce z twardego tworzywa sztucznego. Druk jest możliwy w trybie normalnym (draft) i korespondencyjnym (NLQ). Prędkość druku wynosi 160 znaków na sekundę (character per second — CPS), a w trybie NLQ — 40 CPS. W poprzednim modelu — Amstrad DMP3000 — prędkości te wynosiły odpowiednio 105 i 26 CPS.

Znaki drukowane są z matrycy 9 na 9 punktów. Możliwe jest użycie czterech podstawowych krojów pisma: Pica — 10 znaków na cal (character per inch — CPI), Elite — 12 CPI, Condensed — 17 CPI oraz Proportional — liczba znaków zależy od ich rodzaju. Daje to odpowiednio 80, 96 i 132 znaki w linii. Krój Proportional jest także możliwy do uzyskania w trybie NLQ. Dodatkowo każdy z krojów pisma może być drukowany w podwójnej szerokości (5, 6 i 8,5 CPI). W trybie draft wszystkie kroje pisma mogą być pochylone (italics). Można wybrać odstęp między wierszami 1/6, 1/8 lub 7/72 cala, albo zaprogramować odstęp n/72 lub n/216 cala.

Drukarka jest przystosowana do pracy w dwóch standardach: Epson

FX i IBM Graphics Printer. Wyboru standardu dokonuje się przełącznikiem — programowe przełączenie jest niemożliwe. W trybie Epson drukarka dysponuje zestawem 96 znaków oraz ośmioma zestawami międzynarodowymi po 12 znaków. Znaki o kodach powyżej 127 są wybierane z zestawów podstawowych i drukowane kursywą. Oprócz tego możliwe jest zaprogramowanie przez użytkownika własnego zestawu znaków. W trybie IBM dostępny jest pełny zestaw znaków graficznych standardu IBM (razem 226 znaków). Niemożliwe jest jednak korzystanie z zestawów międzynarodowych i programowanie własnych znaków.

Sterowanie pracą drukarki odbywa się za pomocą przycisków znajdujących się na przedniej ścianie. Dostępne są następujące funkcje: połączenie lub odłączenie drukarki i komputera (ON LINE), przesunięcie papieru o stronę (FF — Form Feed) oraz o wiersz (LF — Line Feed). Dodatkowo włączenie drukarki z wciśniętym przyciskiem LF wywołuje program testujący, a równoczesne wciśnięcie LF i FF przy włączaniu drukarki ustawia ją w trybie kopiowania szesnastkowego. Obok przycisków umieszczone są trzy diody LED sygnalizujące włączenie zasilania, połączenie z komputerem i brak papieru. Na tylnej ścianie, w pobliżu złącza Centronics znajduje



się zespół 18 mikroprzełączników do trwałego ustalania parametrów pracy drukarki.

### EKSPLOATACJA

Drukarka wykorzystywana była do współpracy z różnymi komputerami: IBM PC, Amstrad CPC-6128 i Atari 800XL. Ze wszystkimi komputerami drukarka pracowała poprawnie. Dzięki zgodności z drukarkami Epson FX możliwe jest wykorzystanie „driver'ów” przeznaczonych dla tego standardu. Pozwala to na wydruk wszystkich krojów pisma wraz z dodatkowymi elementami — podkreślanie, wytłuszczenie, pisanie indeksów i wykładników. Jakość pisma jest dobra i przewyższa drukarki Star, ustępując jednak drukarce Triton (zob. „Bajtek” 12/87), która daje wyjątkowo wyraźny wydruk. Także w trybie graficznym drukarka pracowała poprawnie, choć pozostawiała widoczny ślad przejścia głowicy w miejscach bardziej zaczerntonionych. Przykłady wydruków wykonanych na DMP3160 pokazane są obok.

Użytkowanie drukarki jest łatwe i

Test Bajtka — drukarka Amstrad DMP3160  
Test Bajtka — drukarka Amstrad DMP3160  
Test Bajtka — drukarka Amstrad DMP3160  
Test Bajtka — drukarka Amstrad DMP3160  
Test Bajtka — drukarka Amstrad DMP3160



# JAK TO ROBIĄ INNI

## KOMPUTER W ROLI BAGAŻOWEGO

**Czym jest zmora czasami kilkudziesięciominutowego oczekiwania na swój własny bagaż, po wyczerpującym powietrznym rejsie, wiedzą doskonale wszyscy, którzy choć raz znaleźli się w mikroskopijnym w skali największych portów lotniczych świata budynku przylotowym warszawskiego Okęcia.**

Kufry, nesesery, walizki, torby i pakunki trafiają u nas do samolotu i są z niego wyladowywane nadal dzięki pracy oczu i mięśni silnych mężczyźni w charakterystycznych granatowych drelichach. A przecież bez sprawnego i wydajnego systemu opieki nad tym wszystkim, co podróżni, już we wstępnym kontakcie z obsługą lotniska powierzają jej pieczę, można tylko śnić o wielu milionach pasażerów przybywających do Warszawy i odlatujących z niej z uśmiechem zadowolenia. Nim dowiemy się kto i w jaki sposób będzie projektował nasz nowy port lotniczy — Okęcie II — rzućmy okiem na to, jak przy pomocy najnowszych technologii i wszechwładnej elektroniki z zalewem bagażowym radzą sobie inni.

Bazą Królewskich Holenderskich Linii Lotniczych (KLM) jest lotnisko Schiphol w Amsterdamie. Kiedy spogląda się na fotografię z lat trzydziestych trudno sobie wyobrazić, iż Okęcie i Schiphol wystartowały z tego samego poziomu. Dziś amsterdamskie terminale powietrzne zdaniem wielu ekspertów należą do najsprawniej działających na świecie. O skali ruchu na tym jednym z największych i najruchliwszych portów lotniczych globu świadczy, iż w skali rocznej obsłużyć może od 16 do 18 milionów pasażerów i przeladować 650 tys. ton frachtu. Holendrzy jednak liczą się poważnie z rozszerzeniem sieci połączeń KLM, częstotliwości lotów, a także z naporem innych powietrznych towarzystw korzystających ze Schiphol. Wedle ich szacunków przed rokiem 2000 lotnisko należy przystosować do przyjmowania podwójnej liczby pasażerów i podwójnej masy ładunków.

Zdecydowano więc, że do pracy zapręgnie się znacznie bardziej wydajne od ludzi automaty sterowane komputerowo. Ich obwodom elektronicznym powierzy się odpowiedzialność za sortowanie i sprawną transport całych gór bagażu. Cały program rozbudowy Schiphol obliczony jest na sumę 1,4 mld florenów. A sam system bagażowy pochłonie 50 milionów florenów. Ruszą już w roku bieżącym.

Cała maszyna jest pod ziemią pod płytą lotniska i tzw. „palcem C” portu Schiphol. Układ magazynowy i mechaniczny zajmuje powierzchnię 16 tys. metrów kwadratowych, czyli, jak obliczyli skrupulatni Holendrzy, dwóch stadionów piłkarskich. Pozornie rzecz cała sprowadza się do sieci przenośników, jednak jeśli się weźmie pod uwagę skalę ruchu i specyficzne życzenia pasażerów (niektórzy łąpią w Amsterdamie połączenia do różnych zakątków świata, wchodzą i schodzą z pokładów innych samolotów nie należących do KLM) problem nieco się komplikuje.

Wszystko zaczyna się od jednego z 80 stanowisk odprawy bagażowej na Schiphol. Każde zaopatrzone jest w terminal komputerowy i drukarkę nanoszącą dane o podróżnym i jego dobytku, specjalnym kodem na

samoprzylepne kawałeczki folii. Bagaż i bilet pasażera opatrzone są w ten sposób przygotowanymi nalepkami. W tym momencie do dzieła przystępują automaty systemu Baggage Handling System. 1300 laserowych czytników od tej pory kontrolować będzie nie tylko układ linii na nalepkach (taki sam jak w przypadku oznaczeń towarów przeznaczonych do sprzedaży w sklepach posiadających elektroniczne kasy), ale także odpowiedni krój czcionek literowego kodu. Wstępna segregacja waliz następuje na jednym z dwóch przenośników głównych o długości 500 m. Później trafiają na dalsze 250 mniejszych. W sumie 5300 m specjalnych stale przesuwających się taśm napędzanych silnikami o całkowitej mocy 1600 kW.

Przy każdym zakręcie bagażu system sprawdza zgodność wszystkich zawartych na nalepce informacji z danymi centralnego banku pamięci. Nie ogranicza się więc do przeglądu numeru lotu. Mózgiem potrójnej sieci elektronicznej są cztery centralne komputery Microvax umieszczone w odizolowanym od otoczenia podziemnym pomieszczeniu. Przy na zmiany pełniące dyżury i czwarty zapasowy. W przypadku jakiegokolwiek niezgodności „podejrzana” waliza kierowana jest na oddzielny taśmociąg i sprawdzana przez... ludzkie oko i ręce.

Holendrzy szacują, że w ten sposób czas, który pochłania droga bagażu od stanowiska odprawy do luku, czy kontenera bagażowego, skrócony zostanie do 7—8 minut. „Elektroniczni bagażowicy” nie męczą się tak łatwo. Pracować mają po 19 godzin dziennie, 365 dni w roku i są tak zaprogramowani, że praktycznie nie mogą się im zdarzyć pomyłki.

Dla tych ciekawych, którym nie dane będzie osobiście zapoznać się z działaniem komputerów w roli bagażowych przedstawiamy kompletny wydruk na nalepce bagażowej, jaką otrzymałby hipotetyczny Mr Johnson z Atlanty podróżujący liniami KLM z Nowego Jorku przez Amsterdam do Warszawy.

Wojciech Łuczak



wygodne. Przy korzystaniu z różnych rodzajów i formatów papieru ustalenie jego położenia w drukarce jest znacznie ułatwione przez oznaczenia formatów naniesione ponad miejscem wprowadzania papieru. Samo wprowadzanie papieru jest bardzo proste z powodu jego poziomego przechodzenia przez drukarkę. Jednak przy korzystaniu z papieru ciągłego (rolka lub blok) ujawnia się dość uciążliwa wada. DMP3160 w odróżnieniu od drukarek innych firm nie posiada krawędzi umożliwiającej oderwanie zapisanej części papieru.

Przy stosowanym w naszej redakcji systemie drukowania jednego egzemplarza w trybie korespondencyjnym i kopii w trybie draft odczuwalny jest brak na przedniej ściance drukarki przycisku przełączającego te tryby (NLQ — draft). Przycisk taki jest już standardem w drukarkach firmy Star i wielu innych.

Ostatnio daje się zauważyć interesująca tendencja. Każda nowa drukarka — nawet jednej firmy — pojawiająca się na rynku ma inny system prowadzenia taśmy i inną kasetę do taśmy. W DMP3160 zastosowano taśmę o szerokości 6 mm i długości około 70 cm umieszczoną w wyjątkowo nietypowej dwuczęściowej kasecie. Rodzi to obawy, że taśma będzie wymagała częstego nasączania, a ewentualna wymiana może następczą wiele kłopotów.

Praca drukarki jest dość cicha, głównie dzięki osłonięciu głowicy drukującej od strony użytkownika innymi elementami mechanizmów i ich obudową.

Obudowę drukarki wykonano z jasnego tworzywa, a pokrywa jest przyściemniata. Wykonanie jest dokładne, a przyciski i przełączniki pewne w działaniu. Całość urządzenia prezentuje się nowocześnie, choć nie ekstrawagancko. Pewien niepokój może budzić delikatność drukarki, lecz jest to tylko wrażenie spowodowane nietypową konstrukcją.

### PODSUMOWANIE

Drukarka Amstrad DMP3160 nie jest jakąś rewelacją, ale stosunkowo solidnym urządzeniem o przeciętnym standardzie. Posiada wszystkie możliwości drukarek średniej klasy i dzięki dwóm trybom pracy (Epson i IBM) znajdzie niewątpliwie szeroki krąg nabywców. Ze względu na dość krótką i

nietypową taśmę nie polecamy jej jednak do pracy biurowej, w której wymagana jest duża liczba wydruków.

Zalety drukarki:

- proste prowadzenie papieru z wprowadzaniem z przodu
- duża liczba krojów i formatów pisma
- dwa tryby pracy: Epson i IBM
- możliwość definiowania polskich liter
- duża szybkość druku
- łatwe i wygodne wprowadzanie różnych formatów papieru

Wady drukarki:

- nietypowa kasetka na taśmę
- brak krawędzi do odrywania zadrukowanej części papieru z rolki
- brak przełącznika trybów NLQ i draft na przedniej ściance

Wojciech Zientara

### PARAMETRY TECHNICZNE DRUKARKI DMP3160 WG SPECYFIKACJI PRODUCENTA

<b>głowica</b>	— 9-igłowa, możliwość wymiany przez użytkownika
<b>szybkość druku</b>	— 160 znaków na sekundę w trybie zwykłym (draft) 40 znaków na sekundę w trybie korespondencyjnym (NLQ)
<b>matryca znaków</b>	— 9 na 9 punktów (znaki normalne) 9 na 10 punktów (znaki poszerzone)
<b>wielkość znaków</b>	— 2,1 na 2,55 mm
<b>gęstość druku</b>	— 10 znaków na cal (pica) 12 znaków na cal (elite) 17 znaków na cal (condensed) 5 znaków na cal (pica poszerzony) 6 znaków na cal (elite poszerzony) 8,5 znaku na cal (condensed poszerzony)
<b>odstęp wierszy</b>	— 1/6, 1/8, 7/72, n/72 i n/216 cala
<b>papier</b>	— perforowany: 115—254 mm bez perforacji: 100—245 mm
<b>interfejs</b>	— równoległy Centronics (Amphenol) 8-bitowy
<b>zasilanie</b>	— 220—240 V, 50 Hz
<b>wymiary</b>	— długość 400 mm szerokość 250 mm wysokość 100 mm
<b>masa</b>	— 4,2 kg



**WSZYSTKO DLA WSZYSTKICH**

# refleks

**NASZA  
OFERTA!!!**



**ASCOM TECHNOLOGIES**  
(FAR EAST) PTE LTD

## **PWPO-T „Refleks” Sp. z o.o. informuje,**

że działa jako wyłączny przedstawiciel serwisowy na zasadzie zawartego kontraktu z ASCOM TECHNOLOGIES (FAR EAST) PTE LTD. Na zakupiony w tej firmie sprzęt wydawane jest w Polsce świadectwo jakości i udzielana jest roczna gwarancja, w czasie której funkcje gwaranta sprawuje na zasadzie wyłączności PWPO-T „REFLEKS”.

Sprzęt zakupiony w ASCOM po odebraniu przesyłki przez użytkownika jest testowany i sprawdzany bezpłatnie w PWPO-T „Refleks” Sp. z o.o.

### **UŻYTKOWNIK OTRZYMUJE TYLKO DOBRY SPRZĘT!**

Ponadto „Refleks” udzieli Państwu wszelkich dodatkowych informacji zarówno handlowych, jak i technicznych (katalogi, cenniki itp.).

Kontakt: **Przedsiębiorstwo Wdrażania Postępu Organizacyjno-Technicznego „Refleks” Sp. z o.o. Dział Importu, 02-051 Warszawa, ul. Glogera 1 tel. (02) 659-20-41, (02) 659-39-22 tlx 817530 ref pl.**

Wysyłkowo z firmy ASCOM TECHNOLOGIES (FAR EAST) PTE LTD otrzymacie Państwo sprzęt mikrokomputerowy wysokiej jakości i w krótkich terminach dostawy:

#### **Oferta po atrakcyjnych cenach:**

- kompletne zestawy mikrokomputerów PC/XT 6/8/10 MHz, PC/AT 8/10/12 MHz, PC/38612/16/20 MHz oraz inne, jak np. mikrokomputery przenośne i najnowsze typy profesjonalnych mikrokomputerów,
- pełny asortyment kart CSKD, wyposażenia i akcesoriów umożliwiających samodzielne zbudowanie mikrokomputera lub rozszerzenie zestawu już posiadanego (karty główne, grafiki, kontrolery, karty obsługi wejść/wyjść, kable, obudowy, klawiatury, zasilacze),
- pełny asortyment urządzeń zewnętrznych, takich jak: monitory monochromatyczne i kolorowe (szeroka gama typów o różnej rozdzielczości), pamięci taśmowe, pamięci na miękkich dyskach i napędy dysków twarde (o bardzo dużej pojemności i krótkim czasie dostępu), różne typy drukarek firm: EPSON, CITIZEN, STAR, PANASONIC, Amstrad, różne typy ploterów i digitizerów,
- **nośniki magnetyczne,**
- **inne wyposażenie w środki techniki biurowej,**
- **urządzenia i przyrządy elektroniczne,**
- **urządzenia techniki video,**
- **elementy i podzespoły elektroniczne.**

**ASCOM TECHNOLOGIES/FAR EAST/PTE LTD**

**Republic of Singapore**

**45 Genting 05-02 Genting Warehouse Complex Singapore**

**1334 Republic of Singapore.**

**Przedsiębiorstwo Wdrażania Postępu Organizacyjno-Technicznego**



**Sp. z o.o.**

**K-185**

# WSZYSTKO DLA WSZYSTKICH

# Polanglia Ltd

171-175 Uxbridge Road, London W 13 9AA

Tel: London 8401715 Telex: 946581 Polan G Fax: 8407136

**NAJNIŻSZE CENY W EUROPIE ZA NAJLEPSZY SPRZĘT KOMPUTEROWY**  
Wyłączne przedstawicielstwo na POLSKĘ firmy

# AMSTRAD

Rewelacja Roku — Najlepszy PC na rynku

## AMSTRAD PC 1640 ECD

idealny do businessu w pełni zgodny z IBM, maksymalne rozszerzenie skali kolorów [do 64 kolorów], zgodny z EGA, Hercules, MDA i CGA. W składzie mysz, zegar, oba interfejsy i software podobnie jak w PC 1512.

PC 1640 SD MD	Pojedyńcza stacja dysków, monochrome monitor	£ 450.-
PC 1640 SD CD	Pojedyńcza stacja dysków, kolorowy monitor	£ 580.-
PC 1640 SD ECD	Pojedyńcza stacja dysków, kol. monitor wysok. rozdzielczości	£ 700.-
PC 1640 DD MD	Podwójna stacja dysków, monochrome monitor	£ 540.-
PC 1640 DD CD	Podwójna stacja dysków, kolorowy monitor	£ 670.-
PC 1640 DD ECD	Podwójna stacja dysków, kol. monitor wysokiej rozdzielczości	£ 800.-
PC 1640 HD MD	Twardy dysk 20 MB, monochrome monitor	£ 800.-
PC 1640 HD CD	Twardy dysk 20 MB, kolorowy monitor	£ 950.-
PC 1640 HD ECD	Twardy dysk 20 MB, kol. monitor wysokiej rozdzielczości	£ 1080.-

Najpopularniejszy PC Europy:

## AMSTRAD PC 1512

[40% rynku brytyjskiego] — **PO ZNIŻONYCH CENACH:**  
Zgodny z IBM. w skład wchodzi: mysz, zegar Quartz, oba interfejsy, software: MSDOS, DOS+, GEM z Desktop & Paint, Locomotive BASIC 2.

PC 1512 SD MM	Pojedyńcza stacja dysków, monochrome monitor	£ 360.-
PC 1512 DD MM	Podwójna stacja dysków, monochrome monitor	£ 450.-
PC 1512 SD CM	Pojedyńcza stacja dysków, kolorowy monitor	£ 500.-
PC 1512 DD CM	Podwójna stacja dysków, kolorowy monitor	£ 600.-
10-DS	uniwersalne dyskietki 5 1/4" D/S D/D [10 sztuk]	£ 10.-
100-DS	uniwersalne dyskietki 5 1/4" D/S D/D [10 pudełek po 10 szt.]	£ 55.-

\* Polanglia Ltd. dodaje bezpłatnie 10 dyskietek 5 1/4" D/S D/D przy zakupie \* każdego PC 1512 lub PC 1640 oraz książkę i 6 dysków: Migent/Ability + 4 gry wraz z każdym PC 1512

**RABAT : £ 25.-**

przy zakupie PC 1512 lub PC 1640 wraz z rewelacyjną drukarką AMSTRAD DMP 4000 [Drukarka Roku '87], oraz przy zakupie dwóch lub więcej PC na jednego odbiorcę

NAJWYŻSZEJ klasy

## DRUKARKI AMSTRAD

po zadziwiająco niskich cenach [z kablem]:

* Nowa DMP 3160: 160 cps [NLQ 40 cps]	£ 150.-
* DRUKARKA ROKU: DMP 4000 — 15", 200 cps [NLQ 50 cps]	£ 240.-
* 100 różnych możliwości druku włącznie z grafiką	£ 240.-
* LQ 3500 — 160 cps [NLQ 50 cps]	£ 240.-

## AMSTRAD PPC 512/640

W pełni zgodny z IBM; klawiatura odpowiednia do najnowszego standardu PS/2; Stacja/ e dysków 720K 3 1/3".

640 x 200 pixel supertwist LCD, zgodny z CGA i MDA.	
PPC 512 S Pojedyńcza stacja dysków, 720 K	£ 360.-
PPC 640 S Pojedyńcza stacja dysków, 720 K	£ 450.-
PPC 512 D Podwójna stacja dysków, 720 K, wbudowany modem	£ 450.-
PPC 640 D Podwójna stacja dysków, 720 K, wbudowany modem	£ 540.-

Najnowocześniejszy komputer/edytor tekstu z drukarką 'LETTER QUALITY'

## AMSTRAD PCW 9512

cena inauguracyjna: £ 450.-  
w skład wchodzi: drukarka 15" 'daisywheel' o doskonałej jakości druku [LQ], monitor 90 kolumn, 512 K RAM + napęd dysków 1 MB + software: LocoScript 2, LocoSpell [słownik angielski], LocoMail.

Na żądanie klientów wznowiono produkcję niezawodnego komputera z edytorem tekstu:

## AMSTRAD PCW 8256 i PCW 8512 — po nowej niższej cenie

PCW 8256 komputer 256K, pojedyncza stacja dysków, monitor, drukarka, software	£ 280.-
PCW 8512 komputer 512K, podwójna stacja dysków, monitor, drukarka, software	£ 370.-

Popularna seria komputerów domowych

## AMSTRAD CPC 464/6128

po rewelacyjnie niskich cenach:

CPC 464 Z komputer 64K z wbudowanym magnetofonem + zielony monitor	£ 150.-
CPC 464 K komputer 64K z wbudowanym magnetofonem + kolorowy monitor	£ 220.-
CPC 6128 Z komputer 128K z wbudowaną stacją dysków + zielony monitor	£ 220.-
CPC 6128 K komputer 128K z wbudowaną stacją dysków + kolorowy monitor	£ 300.-
10-DK dyskietki 3" [10 sztuk]	£ 25.-
FD-1 + kbl dodatkowa stacja dysków do CPC 6128 z kablem	£ 100.-
RS 232 C serial interface do CPC 6128 + software	£ 60.-
MP-2 modulator TV do CPC 6128	£ 30.-
MP-1 modulator TV do CPC 464	£ 15.-
JY-2 joystick do CPC 464 lub CPC 6128	£ 15.-

## Nowe SINCLAIR SPECTRUM PLUS 2 i PLUS 3

[produkcja pod kontrolą jakości AMSTRADA]:

SP+2 komputer 128K z wbudowanym magnetofonem	£ 100.-
JSJ + s/w joystick i software do SP+2	£ 20.-
SP+3+JSJ komputer 128K z wbudowaną stacją dysków	£ 150.-

\* W ceny wliczone są:

wszelkie koszty dewizowe związane z przesyłką, tzn. koszty F.O.B. w Wielkiej Brytanii, opakowanie, ubezpieczenie na transport do Warszawy, Export Licence, itp.

W Polsce zapewniamy serwis na sprzęt AMSTRAD i STAR jedynie zakupiony w firmie Polanglia Ltd.,

Service Gwarancyjny:

wykonywany jest w Polsce za pośrednictwem T.E — I. REMEX dostępny za dodatkową opłatą £ 30.- doliczoną do zamówienia za każdy komputer AMSTRAD lub system PCW, natomiast £ 15.- za każdą drukarkę, komputery Sinclair i pozostały sprzęt objęty tą ofertą. Serwis pogwarancyjny odpłatny w polskich złotych, dostępny jest dla wszystkich klientów Polanglii

Osoby zakupujące sprzęt AMSTRAD w innych firmach eksportowych lub w sklepach nie są uprawnione do korzystania z serwisu AMSTRADA w Polsce.

Jedynie POLANGLIA LTD jest w stanie zapewnić autoryzowany serwis sprzętu komputerowego AMSTRAD w Polsce



Zgodnie z warunkami aktualnej oferty firmy Polanglia Ltd., niniejszym zamawiam:

..... £ .....

..... £ .....

..... £ .....

PLUS kwota pobierana przez Barclays Bank = £ .....4.-

**RAZEM =** £ .....

Załączam czek lub kserokopie zlecenia bankowego na przelew w/w sumy na konto Nr 70736805 w Barclays Bank PLC, Ealing Broadway Branch (kod 20-27-48), 53 The Broadway, LONDON W 5 5JS, zrealizowanego w dniu ...../...../..... przez bank..... oddział.....

w..... Jednocześnie przyjmuję do wiadomości, że, w wypadku odbioru sprzętu objętego gwarancją zobowiązany jestem do zgłoszenia się do zakładu serwisowego w celu rejestracji sprzętu w terminie 14 dni pod rygorem utraty praw gwarancyjnych.

Wszelkie transakcje podlegają warunkom firmy POLANGLIA opartym na prawie angielskim.

Podpis wpłacającego..... Nazwisko i imię..... Data.....

**NAZWISKO I IMIĘ ODBIORCY** .....

**PEŁNY ADRES** .....

	GIEŁDA „BAJTKA” (tys. zł)	PEWEX BALTONA (USD)	RFN (śred.) (DM)
<b>SINCLAIR</b>			
ZX 81	25	—	39
ZX Spectrum 48 KB	90	115	110-150
ZX Spectrum Plus	130	—	180-230
ZX Spectrum 128 + 2	220	—	250
Drukarka SEIKOSHA GP 50S	85	—	99
TIMEX 2048	130	146	—
Joystick	6	—	4-9

<b>COMMODORE</b>			
C-64	210	219	320
C-128	300	299	480
C-128D	—	—	999
Amiga z monitorem kolorowym	1,2 mln	—	3000
Magnetofon 1531	45	48	30
Stacja dyskietek 1541	200	—	399
Stacja dyskietek 1571	240	299	460
Drukarka GP-500	200	—	149
Dyskietki 5 1/4" (średnia jakość)	0,8-1,5	3,5	0,3-1,5

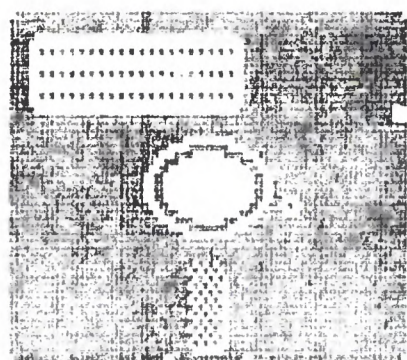
<b>ATARI</b>			
130 XE	100	199	320
65XE	140	125	110
Stacja dyskietek 1050	220	187	350
Drukarka 1029	230	199	270-299
ATARI 520 STM st. dysk. 0.5Mb	1,1 mln	998	800

<b>AMSTRAD</b>			
464 z monitorem monochromat.	280	—	400
6128 z monit. monochromat.	430	—	750
6128 z monitorem kolorowym	530	—	1000
Dyskietki 3"	4-5	—	6-9
Stacja dyskietek 3" do 464	380	—	399
PC 1522	1-2-1,3 mln	—	999

Nieco spokojniejsza styczniowa giełda odnotowała spadek cen i dużą podaż sprzętu. Przede wszystkim oferowano wiele mikrokomputerów z rodziny Spectrum będących pod kontrolą jakościową Amstrada. Pojawiły się już pierwsze egzemplarze Spectrum +3 z trzy calową stacją dysków. Za tę nowość żądano 400 tys. zł.

Natomiast zwolennicy Commodore mieli do dyspozycji kilka mikrokomputerów Amiga 500 i Amiga 1000, ale ich właściciele ograniczali się wyłącznie do wymiany oprogramowania. Obie wersje Amigi wzbudzały ogromne zainteresowanie dzięki rewelacyjnej grafice, przewyższającej rozdzielczością i ilością kolorów nawet najnowszą serię ATARI ST.

(gr)



## INDYWIDUALNY BANK DANYCH

**Fryderyk Badura**, lat 12. Posiada ZX SPECTRUM, kilkanaście gier oraz programów użytkowych. Zainteresowania: filatelistyka i informatyka. Proponuje wymianę programów. Adres: 44-200 Rybnik, ul. Dworcowa 5/41.

**Karol Augustyn**, lat 18. Posiada mikrokomputer AMSTRAD CPC-464. Proponuje wymianę oprogramowania i doświadczeń. Adres: 15-082 Białystok, ul. Nowotki 13/2 m 59.

**Piotr Bilski**, lat 26, lekarz. Mikrokomputer AMIGA 500, drukarka STAR NL-10, stacja dysków 3,5 "AMIGG. Pragnie nawiązać kontakt z użytkownikami amigi w celu wymiany doświadczeń, oprogramowania oraz literatury. Adres: 70-836 Szczecin, ul. Węgierska 5.

**Rafał Gałązka**, uczeń 12 lat. Mikrokomputer Atari 800 XL, stacja dysków 1050. Oprogramowanie: kilkadziesiąt gier i programów użytkowych. Proponuje wymianę gier i programów. Adres: 97-300 Piotrków Tryb. ul. Kostromska 66.

**Mariusz Fredyn**, uczeń 14. Posiada mikrokomputer Commodore 64, magnetofon TURBO DATA. Oprogramowanie: programy użytkowe. Proponuje wymianę doświadczeń i programów. Adres: 03-977 Warszawa, ul. Egipska 5 m 14.

**Rafał Okarma**, uczeń 15. Posiada komputer IBM PC wraz z osprzętem. Oprogramowanie: programy prezentacyjne, użytkowe oraz gry. Pragnie nawiązać korespondencję z posiadaczami komputerów tej marki w celu wymiany doświadczeń, literatury oraz oprogramowania. Adres: 33-171 Pleśnia 284, Tarnów.

**Przemysław Sadoch**, uczeń 13. Posiada Atari 65 XE, joystick, magnetofon XC-12. Posiada programy muzyczne oraz matematyczne. Proponuje wymianę oprogramowania. Adres: 64-920 Piła, Pl. Zwycięstwa 10/11.

**Dudek Grzegorz**, uczeń 15. Posiada mikrokomputer Commodore 16/116. Zainteresowania informatyka oraz elektronika. Proponuje wymianę gier oraz programów użytkowych. Adres: 44-100 Gliwice, ul. Lotników 87/29.

**Ireneusz Niezgoda**, lat 15. Posiada mikrokomputer Commodore Plus 4. Proponuje wymianę oprogramowania. Adres: 05-400 Otwock ul. Batorego 36 m 18.

**Robert i Olgierd Zych** są posiadaczami mikrokomputera Sharp MZ-800. Pragną nawiązać kontakt z kimś kto zna ten mikrokomputer. Adres: 98-400 Wieruszów, ul. Fabryczna 6 bl 1/43.

**Damian Kulak**, uczeń 19. Posiada Atari 800 XL oraz 130 XE. Pragnie nawiązać kontakt w celu wymiany gier i programów użytkowych. Adres: 72-200 Nowogard, ul. Zamkowa 4/1.

**Mariusz Tańczak**, lat 14. Posiada mikrokomputer ZX Spectrum 48 KB. Oprogramowanie około 120 gier, programy własne. Proponuje wymianę oprogramowania oraz literatury. Adres: 60-710 Poznań, ul. Kanałowa 15/17.

# ATARI PROGRAMY POCZTA

DO NAJPOPULARNIEJSZYCH  
PROGRAMÓW DOŁĄCZAMY  
INSTRUKCJE  
NIE CZEKAJ, NIE ZWLEKAJ  
SKORZYSTAJ Z OFERTY  
STUDIO TAL-QWERTY

skr. poczt. 51  
02-105  
WARSZAWA 21

GEMMA STUDIO KOMPUTEROWE

81-969 Gdynia 2

skr. poczt. 149

↑↑ SZBKO ↑↑ TANIO ↑↑ SOLIDNIE ↑↑

AMSTRAD      programy | katalogi  
ATARI        literatura | gratis

## REKLAMUJ SIĘ W BAJTKU

Polski Interface  
Centronics do Spectrum!

- polskie litery na ekranie i drukarce
- druk standardowy i korespondencyjny
- poszerzenie możliwości graficznych komputera do wymiarów 512 x 352 punkty
- kopie rysunów w trzech formatach
- współpraca z drukarkami standardu Epson/IBM

Informacje i zamówienia  
**AKVIS Sp. z o.o.**  
ul. Biskupia 10/14, 31-144 Kraków, tel. 331908

Interface do Spectrum  
ATARI  
drukarki Centronics, RS232 dowolnego magnetofonu, joysticka Sterownik świateł, makiet, reklam, inne. Warszawa, tel. 659-38-44 D-6

Wymieńmy dyski ATARI 1050,  
Stanisław Humela, Częstochowska 37, 98-300 Wieluń. G-4

Taśmy barwiące do drukarek: Commodore 801, Seikosha, GP 500 i 550 A, Atari 1029 (czarne i czerwone) sprzedam. J. Jacheć, 47-330 Zdzieszowice, ul. Dunikowskiego 7, tel. 261 wew. 2357 (całą dobę). K-214

„MIKROFAN” 45-064 Opole,  
skr. poczt. 158 oferuje gry,  
programy użytkowe, opisy na  
ATARI XL/XE. D-10

Kasety magnetofonowe hurtem.  
Wysoka jakość. Tel. 25-57-88  
02-057 Warszawa, Filtrowa 64/45. D-218



NAJWIĘKSZY DYSTRYBUTOR  
PERYFERII KOMPUTEROWYCH  
W RFN

# SYNELEC

Datensysteme  
GmbH  
Postfach 151727  
8000 München 15  
tlx 5212289  
tel 089/519278

## OFERUJE:

Drukarki komputerowe CITIZEN  
120D, MSP-10E, MSP-15E, MSP-40, MSP-45,  
MSP-50 (kolor) MSP-55 (kolor), HQP-45 (24-  
igłowa), Premiere 35 (rozetkowa)

### REWELACYJNE CENY

drukarka CITIZEN typ 120D (10 cali  
NLQ + interface) — 429 DM  
przy zakupie powyżej 10 szt. — 399 DM  
drukarka CITIZEN typ MSP-15E (15 cali  
NLQ + interface) — 668 DM  
przy zakupie powyżej 10 szt. — 618 DM  
kabel do drukarki — 25 DM

## NOWOŚĆ

Drukarka laserowa **UVERTURE 110**  
Koszt transportu jednej drukarki: — 40 DM  
Ponadto:  
— napędy dysków optycznych 115/230 MB  
— streamery ARCHIVE 40—125 MB  
— plottery SEKONIC/SYNELEC  
— terminale ESPRIT — OPUS 220  
— komputery BONDWELL (XT/AT)  
— dyskietki wszystkich typów firmy MAX-  
ELL  
— sieci firmy BICC

**SZYBKA REALIZACJA ZAMÓWIEŃ.**  
— Roczna gwarancja

Autoryzowany serwis na terenie Polski pro-  
wadzi Sp. z o.o. „REFLEKS” 02-051 Warsza-  
wa, ul. Głogera 1

Zamówienia od instytucji:  
METRONEX Sp. z o.o.  
Warszawa, ul. Mysia 2  
tlx 814471

### NINIEJSZYM ZAMAWIAM:

1. .... DM .....

2. .... DM .....

3. .... DM .....

Koszty manipulacji bankowych

DM 10,-

Razem

DM .....

Załączam czek lub kopię zlecenia bankowego na przelew w/w sumy na konto nr 7137320 w Bayerische Vereinsbank München BLZ  
70020270 zrealizowanego w dniu ...../...../..... przez bank ..... oddział ..... w .....

Podpis wpłacającego ..... Nazwisko i imię ..... Data .....

**NAZWISKO I IMIĘ ODBIORCY** .....

**ADRES ODBIORCY** .....

## Cześć Maluchy!

Zgodnie z obietnicą wracamy dzisiaj do naszego nowego przyjaciela, żółwia LOGO. Potrafimy już kierować nim i uczyć go nowych czynności. Spróbujemy więc na początek rysowania kilku figur geometrycznych.

Przyjmujemy jednak pewne założenie — przed rozpoczęciem rysowania żółw powinien znajdować się w dolnej części figury w połowie jej szerokości, po zakończeniu — w górnej. W ten sposób kolejne rysunki umieszczane będą jeden nad drugim.

Rozpoczniemy od zdefiniowania kwadratu:

```
to kwadrat
  rt 90 fd 20
  repeat 3 [lt 90 fd 40]
  lt 90 fd 20
  pu lt 90 fd 40 pd
end
```

Żółw odwraca się w prawo o kąt prosty, maszeruje przed siebie 20 kroków a następnie powtarza (repeat) trzykrotnie tę samą czynność: obrót w lewo o kąt prosty i przejście do przodu o 40 kroków. Mamy już więc trzy całe boki kwadratu i jedną połówkę. Robimy obrót w lewo, dwadzieścia kroków do przodu i kwadrat gotowy. Pozostaje jeszcze sprawa jak umieścić żółwia w górnej części rysunku. Najpierw każemy mu „podnieść pisak” (pu). Od tej chwili za poruszającym się żółwiem nie będzie się już ciągnąć rysowana na ekranie kreska. Teraz obrót w lewo, 40 kroków do przodu i możemy znów „opuścić pisak” (pd).

Skoro potrafiliśmy nauczyć żółwia rysowania kwadratu, z pewnością nie sprawi mu trudności prostokąt:

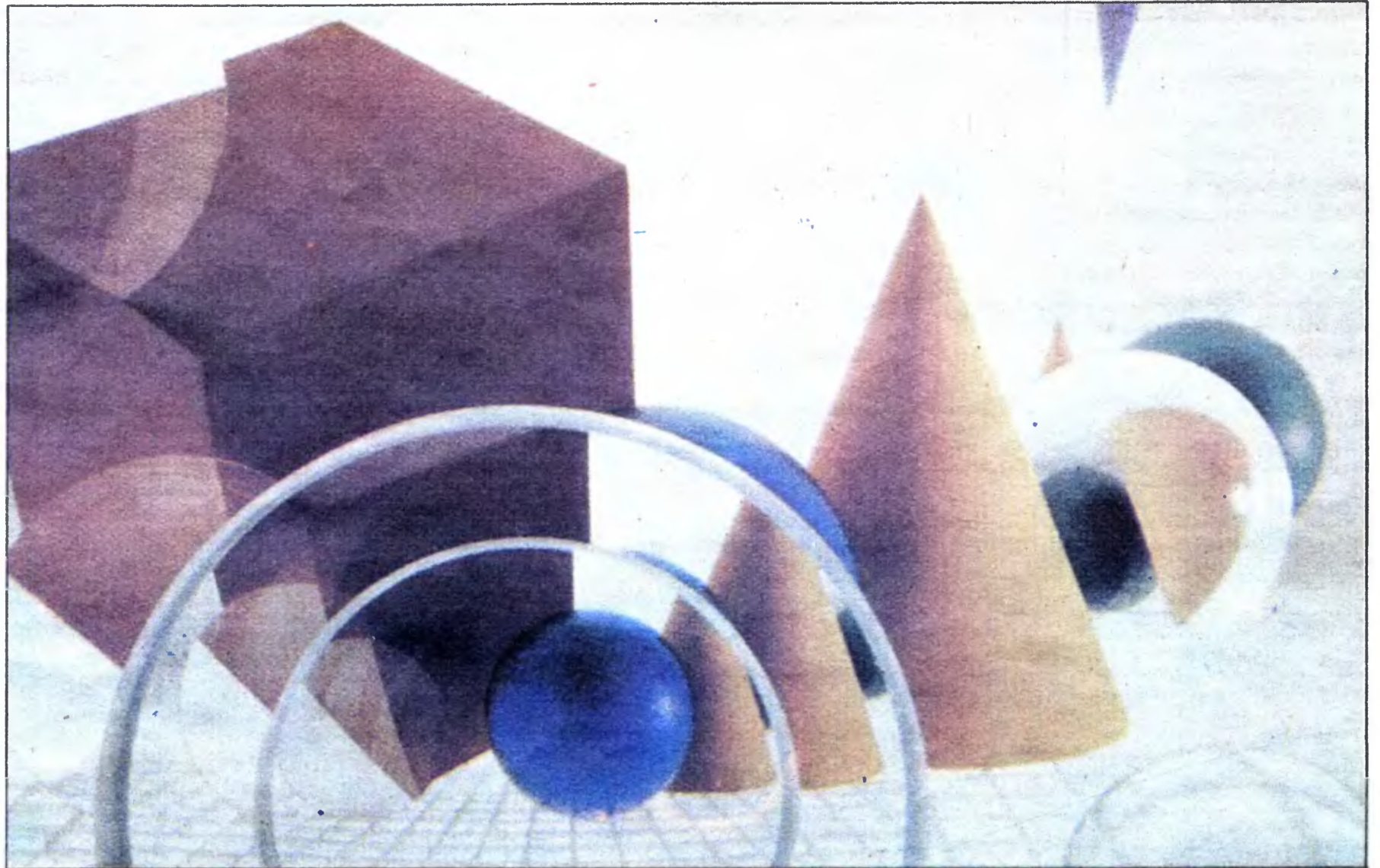
```
to prostokat
  rt 90 fd 20
  lt 90 fd 20
  lt 90 fd 40
  lt 90 fd 20
  lt 90 fd 20
  pu lt 90 fd 20 pd
end
```

a także trójkąt, równoległobok, romb i trapez:

```
to trojkat
  rt 90 fd 20
  repeat 2 [lt 120 fd 40]
  lt 120 fd 20
  lt 90 pu fd 35 pd
end
```

```
to rownolegobok
  rt 90 fd 20
  lt 70 fd 20
  lt 110 fd 40
  lt 70 fd 20
  lt 110 fd 20
  pu lt 90 fd 20 pd
end
```

```
to romb
  rt 60 fd 30
```



# WIEŻE I KOPNIĘTY KWADRAT

```
lt 120 fd 30
lt 60 fd 30
lt 120 fd 30
pu
lt 120 fd 30
pd
end
```

```
to trapez
  rt 90 fd 30
  lt 135 fd 30
  lt 45 fd 20
  lt 60 fd 25
  lt 120 fd 23
  pu
  lt 90
  fd 22
  pd
  end
```

Nieco inaczej podejźmy do rysowania okręgu. Ponieważ żółw LOGO nie potrafi poruszać się po liniach krzywych, narysujemy wielobok foremny (to znaczy taki, który ma wszystkie boki i kąty równe) o takiej liczbie boków, że będzie wyglądał jak okrąg:

```
to okrag
  lt 90
  repeat 60 [fd 2 rt 6]
  pu rt 90 fd 40 pd
  end
```

Nauczmy teraz komputer rysowania wież, które składać się będą z losowo wybranych figur geometrycznych:

```
to wieza.komputera
  pozycja1
  repeat 7
    [make "los random 7
    if :los = 0 [kwadrat]
    if :los = 1 [prostokat]
    if :los = 2 [romb]
    if :los = 3 [rownolegobok]
    if :los = 4 [trapez]
    if :los = 5 [okrag]
    if :los = 6 [trojkat]]
  end
```

polecenie: make "los random 7 oznacza, że zmienna "los" przyjmuje wartość losową spośród liczb całkowitych 0—6. Następujące po tym instrukcje warunkowe tak bardzo przypominają BASIC, że nie trzeba już chyba nikomu tłumaczyć jaką spełniają funkcję.

Pozostaje jeszcze określić co to jest pozycja 1:

```
to pozycja1
  pu
  home
  lt 90 fd 200
  lt 90 fd 100
  lt 180
  pd
  end
```

No i proszę! Z luźnych klocków powstała nam zabawa w budowanie wież. Najpierw komputer buduje swo-

ją wieżę, a następnie my staramy się stworzyć drugą, identyczną z pierwszą. Robimy to wpisując nazwy kolejnych figur geometrycznych, z których składała się wieża komputera. Przyda się to z pewnością tym, którzy jeszcze ciągle myślą trapez z równoległobokiem i nie mogą sobie przypomnieć jak na serio nazywa się „kopnięty kwadrat”.

Najpierw jednak trzeba przenieść żółwia w inne miejsce:

```
to pozycja2
  pu
  home
  rt 90 fd 200
  rt 90 fd 100
  rt 180
  pd
  end
```

Grę będziemy rozpoczynać od wpisania słowa wieze.

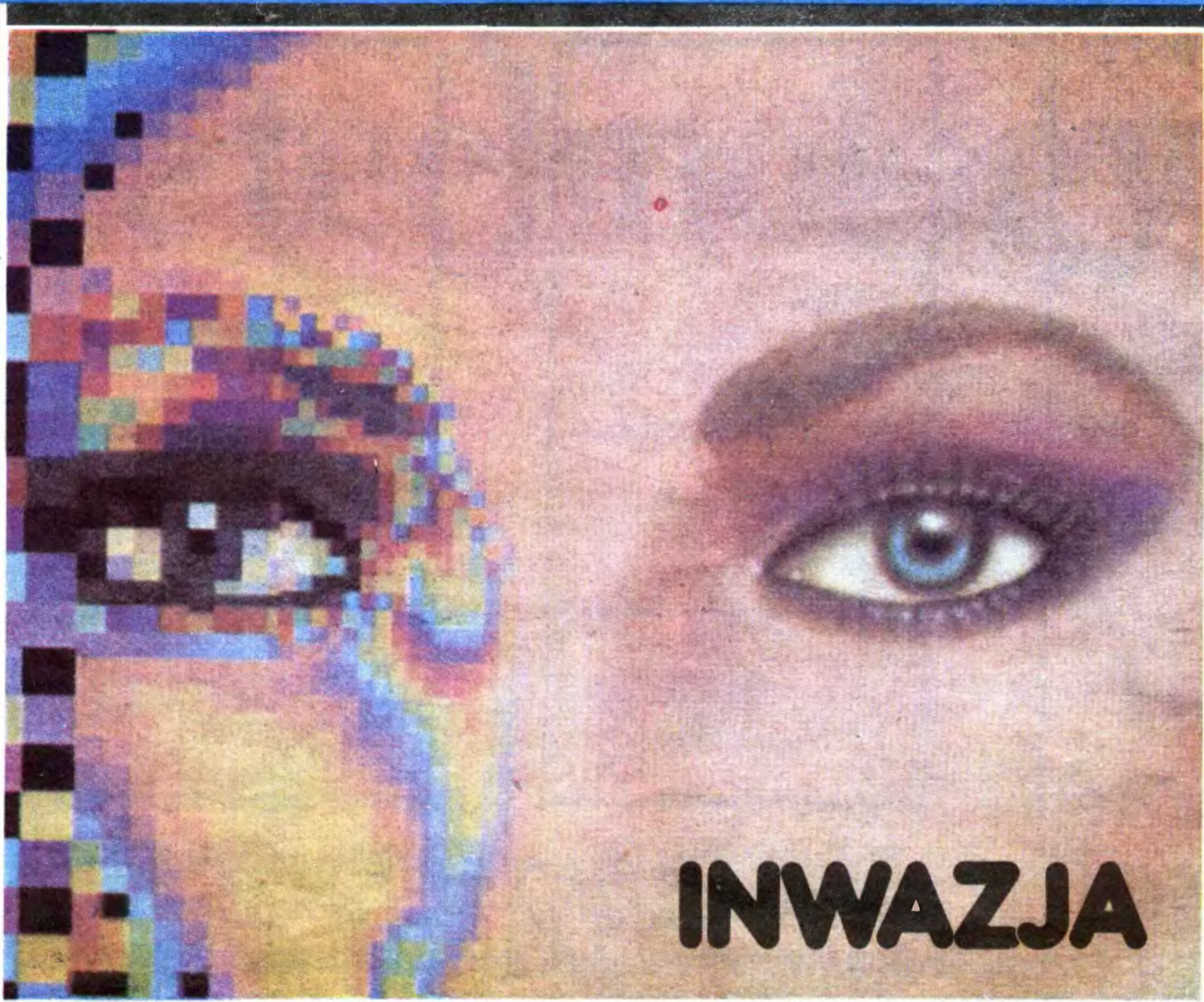
```
to wieze
  cs st
  wieza.komputera
  pozycja2
  end
```

Jeśli ktoś z was nie wie jeszcze co to jest „kopnięty kwadrat”, to niech weźmie cztery listewki jednakowej długości, zbije je gwoździkami tak, by powstał kwadrat i... kopnie tę konstrukcję. Jeśli listewki nie połamią się i wytrzymają gwoździe, to figura którą zobaczycie będzie właśnie „kopniętym kwadratem”.

Nie zniszczcie przy okazji trampek!

Romek

# NIE TYLKO KOMPUTERY



## INWAZJA

# WIDEOFONÓW

Żadna szanująca się rekwizytornia choćby najtańszego i realizowanego najprostszymi środkami filmu science fiction nie może się obyć bez tego jedyne go specyficznego sprzętu. Jego obecność w kadrze symbolizuje to, że ludzkość pokonała już dawno pewien etap w historii komunikowania się na odległość i stał się on czymś zupełnie naturalnym w codziennym życiu przyszłych pokoleń.

Marzenia o wideofonie, czyli aparacie stojącym na znacznie wyższym stopniu rozwoju technologicznego niż zwykły telefon, o systemie przekazyującym rozmówcy nie tylko dźwięki, ale również wizerunek tego z kim się porozumiewa, są chyba jeszcze starsze niż wspaniały wynalazek Grahama Bella. Cóż dziś w tym nadzwyczajnego — powiedzą niektórzy — przecież za pośrednictwem sieci telewizyjnej i satelitów możemy uczestniczyć bezpośrednio w wydarzeniach dziejących się tysiące kilometrów od nas, na innych kontynentach. Nawet my w kraju zdystansowanym wyraźnie przez postęp techniki śledziliśmy całkiem na żywo pierwsze godziny szczytu waszyngtońskiego Gorbaczow — Reagan. Bardziej zorientowani dorzucają jeszcze kilka przykładów — telemosty, specjalne międzykontynentalne konferencje prasowe, prywatne narady satelitarne grubych ryb świata międzynarodowego biznesu. To wszystko niesłychanie oszczędza czas dziennikarzy, polityków, czy ludzi interesu, którzy dążąc na odległe miejsce spotkania, musieliby spędzić w kabinach bardziej lub mniej luksusowo wyposażonych samolotów. Nie oszczędza wszakże pieniędzy. Kilkogodzinne połączenie telewizyjne osób komunikujących się za pośrednictwem pojazdu zawieszono-

nego nad Ziemią pochłania bająskie sumy. Nie możemy więc mówić o powszechności sprawy.

Ową powszechność uzyskamy, kiedy aparat o wymiarach obecnego telefonu podłączony do normalnego przewodu abonenckiego przekaże nam po wybraniu numeru i głos i obraz osoby, z którą pragnęlibyśmy zamienić parę słów, a może pokazać także interesujące ją fotografie, czy też rysunki. Ten wyścig do powszechności wideofonu już się rozpoczął w Stanach Zjednoczonych i Japonii. I to pomimo faktu, że używane w tych krajach linie, zdaniem specjalistów nie bardzo się do tego nadają. A więc trzeba było się dostosować do istniejących sieci.

LU-1000 zamówić może każda firma amerykańska. Firma, bo jest to w tej chwili najdoskonalszy i najdroższy aparat wideofoniczny Ameryki. Kosztuje 1095 dolarów i działa podłączony do najzupełniej normalnego przewodu pod warunkiem, iż pragniemy porozumieć się za jego pomocą z kimś, kto dysponuje drugim LU-1000 zarejestrowanym pod zwykłym numerem telefonicznym. Aparat firmuje usadowiony w Cypress w Kalifornii amerykański oddział słynnego japońskiego konglomeratu Mitsubishi, a sprawą rozwoju wideofonicznej komunikacji zajmuje się świeżo powołana komór-

ka zaufania — Visual Telecom Division. Produkcję powierzono należącej do Mitsubishi wytwórni Luma Telecom, stąd nazwa urządzenia LU-1000.

Wideofon Luma składa się z czegoś w rodzaju komputerowej klawiatury, w którą wbudowano głośniki i mikrofony oraz części wizualnej, czyli minikamery video, a także czarnobiałego ekranu telewizyjnego o przekątnej trzech cali. Połączenie następuje poprzez wystukanie nazwiska osoby, z którą pragniemy rozmawiać. LU-1000 zaprogramowany pamięta dane i numery telefoniczne 100 naszych przyjaciół. Mało tego, na żądanie właściciela wyświetla je na swym monitorze.

Kiedy „podniesiemy słuchawkę” aparat pokazuje jedynie ruchomy wizerunek posługującego się nim człowieka, po uzyskaniu kontaktu z drugim abonentem obraz dzieli się na dwoje. Po jednej stronie widzimy osobę z nami dyskutującą, a raczej zapamiętaną i przesłaną nam przez LU-1000 stopką z portretem tego, kto podszedł do aparatu, po drugiej w pełni ruchomą własną podobiznę. Tak samo nieruchomo odbierani jesteśmy przez naszego interlokutora. To oczywiście niedoskonałość tego pierwszego, powszechnego wideofonu. Ale jego twórcy całą winę zwalają na „zbyt wąskie gardło” w postaci miedzianych przewodów telefonicznych. Jeśli zastąpiłoby się je światłowodami transmisja byłaby absolutnie pełna i żywa, choć nadal w czerni i bieli. Na razie, bo wedle zapewnień producenta to dopiero wprowadzenie do tematu inwazja wideofonów.

Wrażenie ruchu w obrazie naszego rozmówcy w LU-1000 tworzone jest w ten sposób, iż co kilka sekund w zależności od położenia jego twarzy pojawia się nowa stopka. Oczywiście można kamerze podsunąć w ten spo-

sób różne przedmioty, pokazać obrazy, plany, czy szkice. To już bardzo wiele jak na początek.

LU-1000 używany jest obecnie w USA przez kilkanaście wielkich przedsiębiorstw, a także przez niektóre oddziały policji. W tej ostatniej roli aparat oddaje nieocenione usługi jako błyskawiczny identyfikator poszukiwanych, zatrzymanych i podejrzanych o przestępstwo.

Ale amerykańska firma Mitsubishi pomyślała również o tych, których nie stać na skomplikowany wideofon za ponad tysiąc dolarów. Dla nich wprowadzono na rynek również fabrykowany przez Luma Telecom tak zwany konsumencki model taniego i powszechnego aparatu. Nosi nazwę Visual Telephone Display, w skrócie Visitel. Od początku 1988 roku nabyć go można w każdym sklepie prowadzącym tego rodzaju rzeczy, wraz z instrukcją podłączania do gniazdka, za 399 dolarów.

Visitel wyposażony jest w normalną słuchawkę z mikrofonem, lecz nie pamięta numerów i nazwisk tych, z którymi chętnie rozmawiamy, ma bowiem zwyczajny klawiszowy wybierak. Oczywiście dochodzi do tego zmniejszona kamera i ekran. Ten ostatni większy niż w przypadku LU-1000 — czteroipółcalowy. Odbieramy na nim wyłącznie „zamrożony” obraz tego co zobaczy w pierwszym momencie połączenia kamerą Visitel, z którym się kontaktujemy.

Tim Beck — kierownik sieci sprzedaży spółki Mitsubishi America zapewnia wszystkich potencjalnych klientów, że Visitel nie będzie opróżniał kieszeni swoich posiadaczy. Opłaty za wideofoniczną rozmowę mają być podobno identyczne, jak w przypadku zwykłego „ślepego” połączenia. Nie potrzebna jest żadna przystawka, czy konwertytor. Tylko sprawne gniazdko sieci telefonicznej.

Z najnowszych informacji wynika, że tacy potentaci branży elektronicznej jak NEC i Panasonic już kupili licencje Mitsubishi i w ciągu 1988 roku zaprezentują własne aparaty wideofoniczne przeznaczone na rynek amerykański. W Japonii natomiast Sony i Nippon Telegraph and Telephone Corp. (NTT) — największa firma telekomunikacyjna na świecie, wspólnymi siłami starają się przeskoczyć dokonania konkurentów. Ich wideofony, ponoć znacznie doskonalsze, zarówno te dla świata interesu, jak i dla każdego abonenta, spodziewane są już w najbliższych miesiącach w japońskich sklepach.

To wszystko zapowiada kolejny załew. Teraz na progu ery powszechnego wideofonu pozostało pokonanie jeszcze jednej bariery. Torowanie drogi ku temu, co niedawno występowało jedynie w filmach fantastyczno-naukowych hamuje miedziany, błyszczący pręt obłany plastikową izolacją. Giganci branży telekomunikacyjnej proponują najbardziej uprzedmiotowemu krajom Zachodu 10-letni program przejścia do światłowodów. Projektowany układ oznaczono jako Integrated Services Digital Network (Cyfrowy System Zintegrowanych Usług). W ten sposób wszystko — tekst, rysunki, fotografie barwne i obrazy przesyłane będą mogły być łączami abonenckimi 30 razy szybciej niż obecnie. Mówi się, iż już niedługo będzie można zostawić na ekranie własnego wideofonu odręcznie malowaną kilkoma kolorami wiadomość przeznaczoną tylko dla tego jedyne go, który może zadzwonić podczas naszej nieobecności. Na coś miło się o tym pisze czekając w co najmniej kilkuletniej kolejce na przydział swojego numeru i założenie takiego zwykłego, najzwyklejszego telefonu.

Wojciech Łuczak