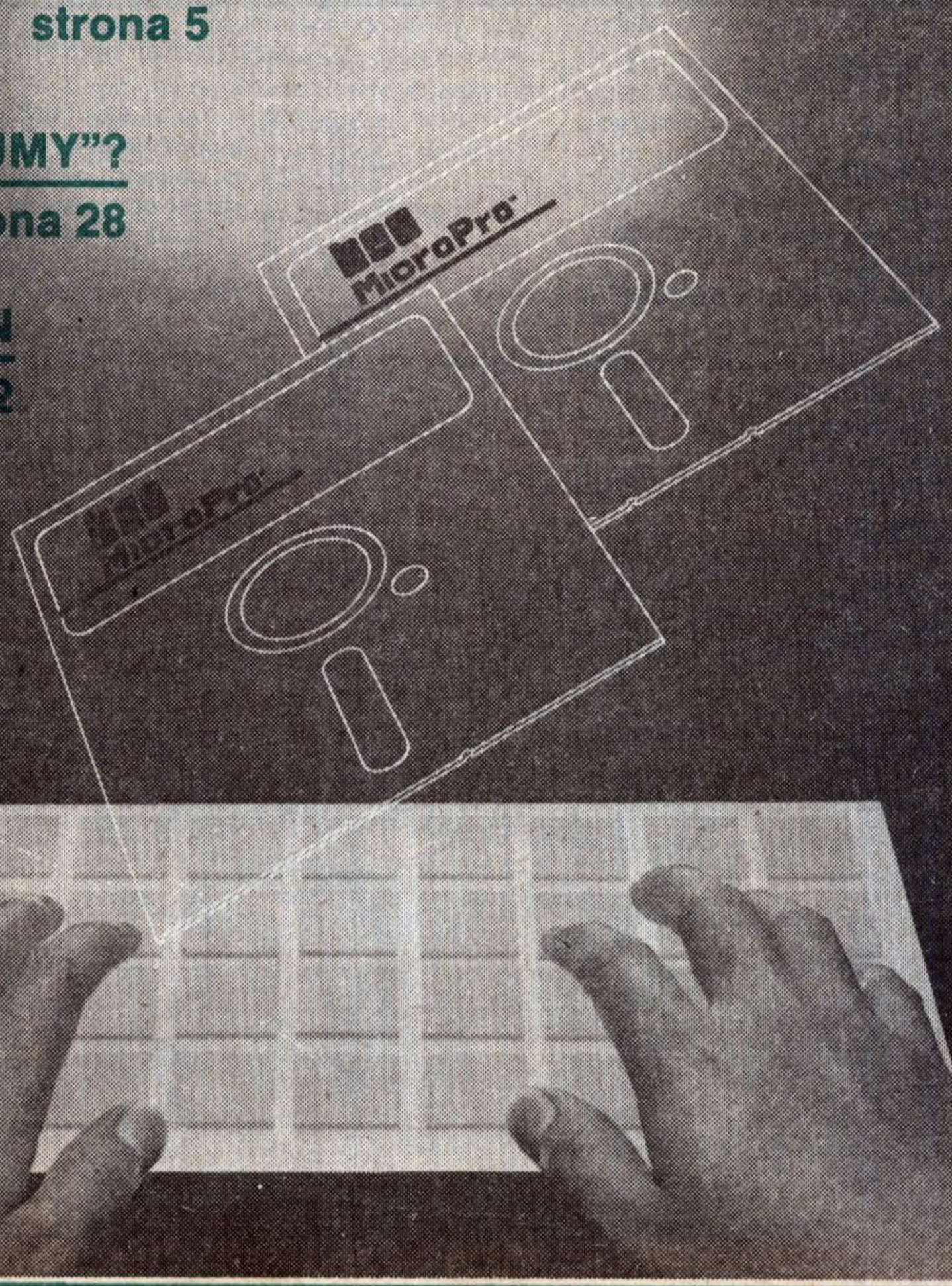
IKS

INFORMATYKA KOMPUTERY SYSTEMY

CENA 120 zł

DODATEK DO „ŻOŁNIERZA WOLNOŚCI" NR 2/1989 ISSN 0860—2794



Od zakończenia drugiej edycji „Ligi Myślących" upłynęło ponad pół roku. Przypominamy: pierwsze miejsce zajął **Zdzisław Bedynek,** *pozostałymi laureatami byli* **Wiesław Rychlicki, Magdalena Rojek, Andrzej Barcewicz i Adam Dominski.**
Zwycięzcy odebrali nagrody z rąk zastępcy redaktora naczelnego „Żołnierza Wolności", płk. Marka Zdziecha.

SPIS TREŚCI

Ten numer „IKS-a" oddajemy do rąk Czytelnika z opóźnieniem. Przepraszamy. Mamy nadzieję, że w obliczu „uzdrawiania gospodarki" i stawiania spraw oczywistych „na nogi" terminowość ukazywania się naszego pisma będzie ekonomiczną koniecznością, prawa rynku odgrywają również w naszej redakcyjnej pracy niemałą rolę.

Sądzimy jednak, że nasze pismo nadal towarzyszyć będzie Czytelnikom poszukującym ciągle nowej wiedzy o informatyce. Jaki będzie „IKS"? O tym decydują jak zwykle Czytelnicy. Uważnie czytamy Wasze listy, choć nie na wszystkie jesteśmy w stanie odpowiedzieć — ale każdą uwagę skierowaną pod naszym adresem starannie wykorzystujemy. Zatem czekamy na Wasze uwagi.

FIRMA	MODEL	CPU	FPU	MIPS	RAM (Mb)	max rozszerzenie RAM (Mb)	HARD DYSKI	MONITOR	Rozdzielczość	System operacji (OS)
APOLLO--COMPUTERS	DN-3000 DN-4000	68020-12 68030-33	68881-12 68881-33	1.3 4	4 8	8 32 lub 64	80/155/330 155/330/696	15"mono 19"mono 19"color 19"color	1024×800 1024—1024 1024×1024 1024×1024	UNIX--Berkeley w.4.3 Domain Aegis
	DN-10.000	RISC	—	12	64	128 lub 256	696	19"color 19"color	1280×1024	
SUN-Microsy-stems	Sun-3/50 Sun-3/160 Sun-4/260	68020-15 68020-25 RISC	68881-16 68881-25 weitek 1164/1165	1.5 4 10	4 8 32	4 16 128	49 141 560	19"mono 16"color 19"color	1152×900 1152×900 1152×900	Sun (OS) (UNIX)
Televideo-systems	15 DL 17 C 19M 19C	80386-16 80386-16 80386-16 80386-16	80287-10 80387-16 80387-16 80387-16	2.2 2.2 2.2 2.2	4 4 4 4	16 16 16 16	— 40 71 71	15"mono 17"color 19"mono 19"color	1280×1024 1024×1024 1024×1024 1024×1024	Microports DOS-Merge -386 (UNIX)
Hewlett-Packard	9000-350 9000-8255	68020-25 RISC	68881-25 —	3.7 8	8 8	32 56	80 80/131	19"color 19"color	1280×1024 1280×1024	UP-Ux (UNIX)
DEC	VAX-Station 2000	II procesor	— —	0.9	4	6	159	15"mono 15"color	1024×864 1024×864	ULTRIX (UNIX) VMS
XEROX	XEROX 228S	68020-16	68881-16	2.5	4	16	86	19"color 19"mono	1024×792 1024×792	Berkeley UNIX 4.3
MASSCOMP	MC 5350 MC 5450	68020-16 68020-20	68881-16 68881-20	2 2.5	2 2	4 10	71 71	12"color 13"mono	640×480 1152×910	RTV (UNIX)

WORK STATION
— STACJE GRAFICZNE
do zastosowań inżynierskich

Już w roku 1980 użytkownicy komputerów; technicy, inżynierowie, naukowcy, architekci i różnego rodzaju projektanci — rozważali dwa ponure fakty: niewielką moc mikrokomputerów i bardzo wysokie ich ceny. Mikrokomputery posiadały niewystarczającą moc jednostek centralnych (CPU), pamięci wspólnych zasobów, jak również nie posiadały właściwej (odpowiedniej dla celów inżynierskich) rozdzielczości ekranów. Te cechy posiadały minikomputery, które z kolei nie umożliwiały interaktywnej pracy, ze względu na swoiste systemy operacyjne. Problemy te rozwiązała najpierw firma APOLLO COMPUTERS w 1981 roku komputerem DOMAIN DN 100, później firma SUN MICROSYSTEMS w 1982 roku komputerem SUN-1 zapełniając lukę Stacjami Roboczymi ogólnego przeznaczenia. Systemy te zostały stworzone dla potrzeb inżynierskich i posiadały szybkie jednostki centralne (CPU), duże pamięci RAM, ekrany o wysokiej rozdzielczości, duże pamięci dyskowe oraz możliwości pracy w sieciach przy użyciu *file serverów*, jak również pozwalały połączyć ze sobą różne stacje z różnymi urządzeniami peryferyjnymi, takimi jak plottery, digitizery, scanery, drukarki itp.

W szczególności APOLLO i kroczący tuż za nim SUN rozrastały się gwałtownie tworząc nowe urządzenia dla elektroniki, projektowania kosmicznego i lotniczego, CAD-u, prac wydawniczych, sztucznej inteligencji, wspomagania komputerowego prac inżynierskich, przetwarzania obrazów, trójwymiarowego modelowania struktur, jak również obsługi służb finansowych i księgowych. Tym samym wzrosło zainteresowanie innych firm komputerowych tego typu produktami.

STANDARD STACJI ROBOCZEJ (GRAFICZNEJ)

Najprostsze stacje robocze są zadziwiająco podobne do siebie, a to za sprawą standardu preferowanego od początku swej działalności przez firmę APOLLO.

Podstawowy typ Stacji Roboczej posiada jednostkę centralną (CPU) 32-bitową, 4 do 8 Mb RAM, 40 do 80 Mb Karol dysk, ekran megapixelowy, połączenia sieciowe o szybkości transmisji rzędu 10 Megaboadów, system operacyjny wielozadaniowy oparty na pamięci Virtualnej (tzw. wspomaganie herolwerowe grafiki). Większość z nich może pracować w sieci we współpracy z file serverem. Zarówno prekursor w dziedzinie produkcji stacji graficznych — firma APOLLO, jak również SUN, HEWLETT PACKARD i prawie wszystkie współczesne stacje graficzne oparte są na Motoroli serw 68000 i koprocesorze 68881 z paroma wyjątkami, jak DEC VAX Stadion 2000. Wszystkie firmy pragną maksymalnie zwiększyć częstotliwość zegara, jednostki centralnej, stosując następny w generacji mikroprocesor. Wreszcie wielu projektantów jednostek centralnych zwraca się do własnych chipów, często opartych na technologii RISC (*Reduced Instruction set Computer)* — bezwzględnym liderem jest APOLLO z modelem DN 10000, którego wartość porównywalna jest z dużymi superkomputerami CRAY-a.

Komputery PC często mierzą swoje osiągnięcia szybkością zegarów. Stacje graficzne natomiast opierają swoje osiągnięcia na milionach operacji na sekundę (MIPS), od 1 MIPS-a w pierwszych modelach do MIPS-ów np. w APOLLO serii 4000, którego cena w rozbudowanej konfiguracji 32 Mb RAM, 330 Mb nad dyskiem i kolorowym monitorem 19 o rozdzielczości 1024×1024 wynosi około 115 000 dolarów, co wydaje się w porównaniu z innymi minikomputerami dość niewygórowaną ceną.

Wszystkie stacje robocze zapewniają wielozadaniowość poprzez system operacyjny i dzięki odpowiedniej architekturze wewnętrznej gwarantują doskonałe osiągnięcia. Większość nowych systemów posiada na przykład pamięć typu Cache Memory (8 Kb), która stała się standardem. Wiele programów inżynierskich

jest bardzo rozbudowanych i wykorzystują często interaktywną grafikę. Inżynierowie często korzystają z wielozadaniowości dzięki czemu mogą przez cały czas pracować, podczas gdy długie i uciążliwe obliczenia dokonywane są w tle.

Cechą stacji, decydującą o sukcesie są duże ekrany z dobrą rozdzielczością mierzoną już w megapixelach, pozwalające praktycznie na stosowanie wielu okien jednocześnie, np. w modelu APOLLO serii 4000 rozdzielczość ekranu 19" wynosi 1280×1024, co jest obecnie absolutnym rekordem.

OPROGRAMOWANIE

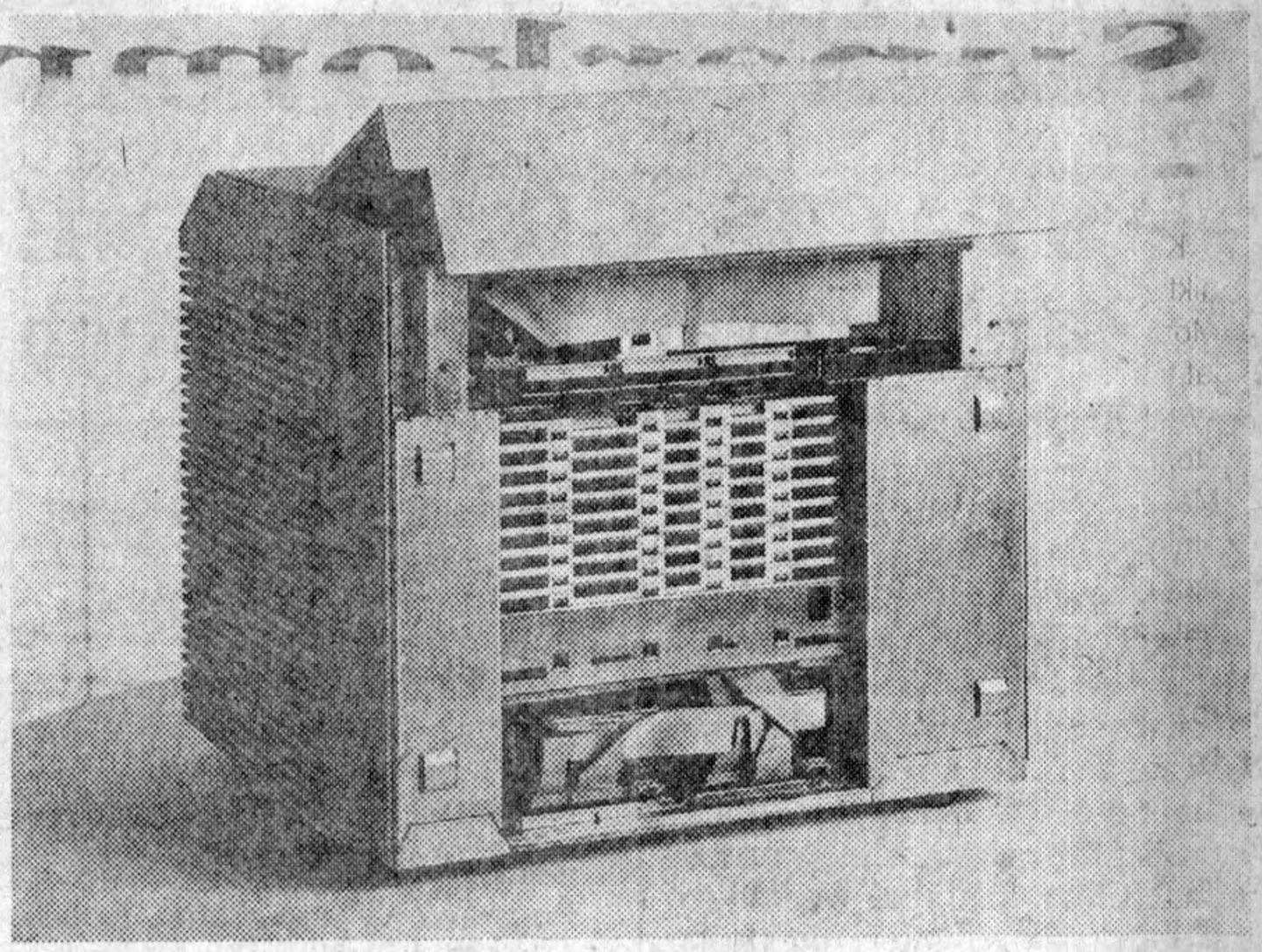
Wspólną cechą wszystkich niemalże Stacji Graficznych jest ich system operacyjny (OŚ). Prawie wszystkie wykorzystują adaptację UNIX — a łączącą osiągnięcia systemu V z elementami systemu BERUELEY 4.2.

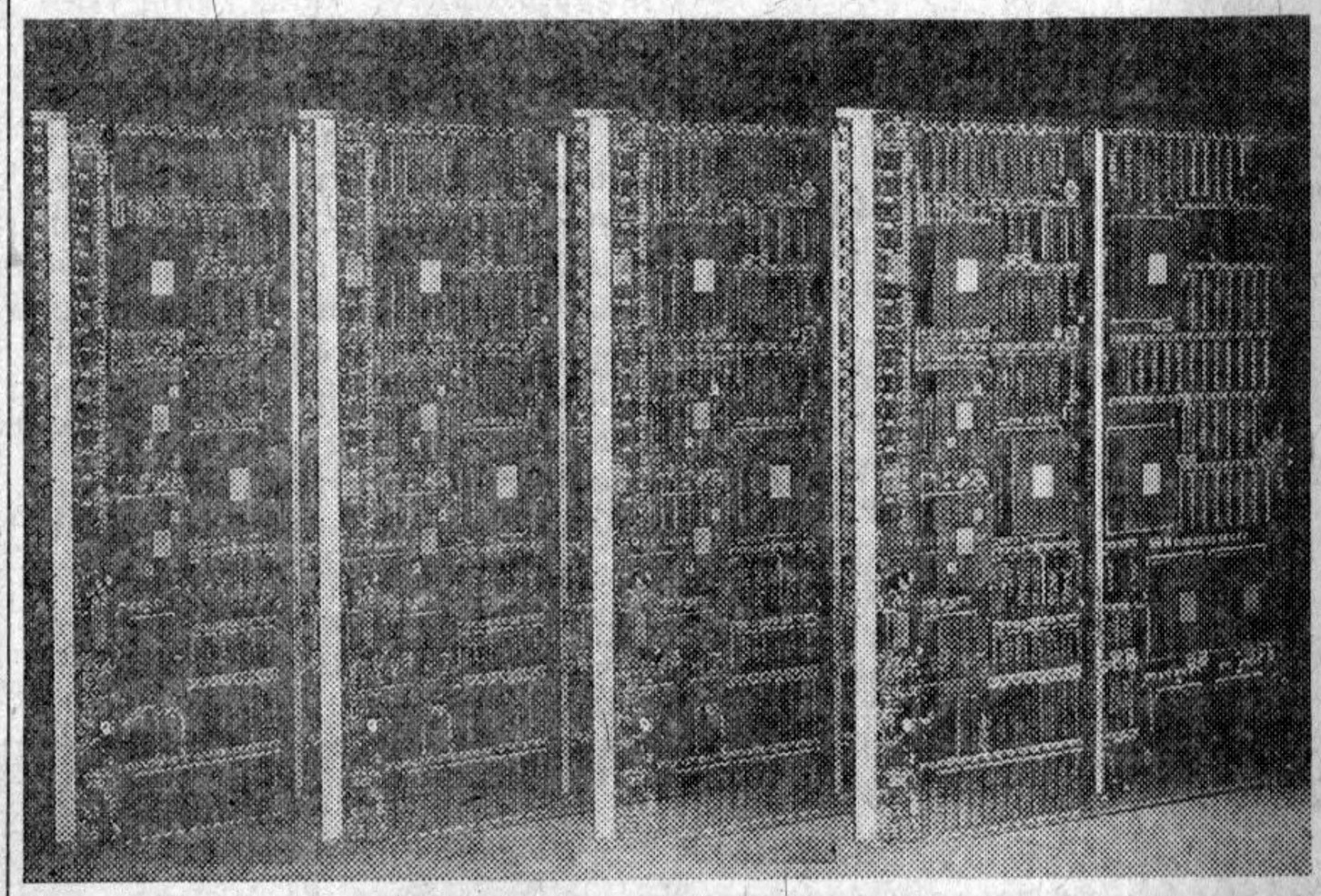
APOLLO rozpoczął pracę z własnym systemem o nazwie AEGIS, ale wkrótce przeszedł na UNIX-a pociągając za sobą prawie wszystkie firmy. SUN krocząc tuż za APOLLO musiał nawet zatrudnić bezpośredniego projektanta systemu UNIX 4.2., myśląc, że to właśnie pomoże mu w dotrzymaniu kroku APOLLO.

Firma DEC oferuje zarówno UNIX, jak i swój własny system operacyjny pod nazwą VMS, który z uwagi na długotrwałą obecność firmy na rynku prawie stał się standardem.

Nowe firmy w branży; jak APPLE, czy Televideo, zdając sobie sprawę z popularności UNIX-a oferują go równolegle z własnym systemem.

Większość użytkowników MACINTOSHY zaraz poczuje się jak w domu, pracując na stacjach roboczych, bo używają one okien do zarządzania systemem. Dwa systemy walczą tu o rangę standardu. APOLLO i jego naśladowcy DEC, HP i wielu innych popiera X-WINDOWS SYSTEM, pakiet stworzony na uniwersytecie MIT jako część projektu ATHENA oraz NEWS (Network-extensible windowing system) preferowany przez firmę SUN, który odstrasza potencjalnych klientów ze względu na zbyt trudny aparat użycia procedur nie znajdujących miejsca przy pracy w sieci. Szybka zintegrowana sieć jest elementem vitalnym, jeżeli chce się wymieniać informacje graficzne pomiędzy użytkownikami, wspólnie korzystać z drogich peryferii, jak plottery, drukarki, duże dyski laserowe czy streamery. Najczęściej używaną siecią w świecie stacji graficznych jest Ethernet. Z uwagi na rolę sieci wielu producentów stacji wytwarza również servery, które są wyspecjalizowane w obsłudze węzłów sieciowych.





Moim celem było przybliżenie problemu mało znanego i niedocenionego na polskim rynku komputerowym, a przecież trudno sobie wyobrazić w wysoko uprzemysłowionych krajach świata prace projektowo-graficzne bez WORUK STATION.

Można optymistycznie przewidzieć wielkość postępu naukowo-badawczego w dziedzinie projektowania i grafiki w Polsce przy zwiększonych wysiłkach popularyzujących ten kierunek informatyki.

BRYSKOW

Superkomputery

Komputer w Uniwersytecie Illinois symuluje coś czego nikt wcześniej nie widział. Pierwsze chwile po „Big Bang'u". Można oglądać wydarzenia, które miały miejsce miliardy lat temu — fantastyczne kształty kolorowych obłoków masy międzygalaktycznej. Owe kolory wskazują różnice gęstości wędrujących chmur materii. Natomiast w Instytucie Massachusetts inny komputer usiłuje nauczyć się tego, co umie trzyletnie dziecko. Na przykład zdolności oceny różnicy między kubkiem a spodkiem. To, co dzieci dostrzegają natychmiast, komputer musi krok po kroku przeanalizować. Najpierw rozpoznać klasę obiektu, fizyczne cechy różniące go od otoczenia, potem zwraca uwagę na atrybuty, itp.

Eksperymenty te wskazują na paradoks istoty współczesnej informatyki. Z jednej strony gigantyczne moce obliczeniowe, z drugiej natomiast intelektualne zdolności, które często są znacznie mniejsze od poziomu intelektualnego kilkuletniego dziecka. Wyraźnie dominują typowe (tradycyjne) zastosowania komputerów — szczególnie tam, gdzie mamy do czynienia ze specjalizowanymi zadaniami. Mimo ich ogromnej skali, maszyny wykonują te prace w czasie kilku godzin — przed erą komputerów wymagało by to zaangażowania wielu ludzi w czasie kilku pokoleń...

Od 40 lat naukowcy pracują nad tymi obiema drogami badań komputerowych. Jedna grupa zajmuje się szybkimi jak światło maszynami — i niezmiennie dąży do zwiększenia ich szybkości, mocy. Druga grupa pisze programy o znamionach początków sztucznej inteligencji. Obie te grupy wybitnych uczonych, dysponując miliardami dolarów działają indywidualnie, jak gdyby druga grupa nie istniała.

Ostatnio pojawiły się pewne oznaki, iż dwie szerokie drogi badań komputerowych mogą się połączyć. Być może wkrótce najlepsze komputery zostaną sprzężone w jeden elektroniczny mózg, który będzie nie tylko nadzwyczaj szybki, ale i mądry. Trudności owego „spotkania" należy głównie upatrywać w różnych poziomach zaawansowania tych prac. Sztuczna inteligencja właśnie rozpoczyna swoje życie. Pierwszy komercyjny projekt powstał zaledwie pięć lat temu. Mimo to ma on już dzisiaj wiele poważnych zastosowań. Jednak z drugiej strony fabryki produkujące sprzęt, o którym jest mowa, czyli superkomputery, zajmują się tym już od ćwierć wieku. Mają ogromne doświadczenia i sprawdzony rynek. Sprzedaż tych maszyn, a kosztują one od 5 do 25 milionów dolarów każda, rośnie o 25% każdego roku.

Około 300 superkomputerów wykorzystywane jest do badań nad problemami paliwowymi, do analizowania struktury mięśni i tworzenia specjalnych efektów dla hollywoodzkich filmów. Ogromna moc superkomputerów dostępna jest, via telefon i sputnik, dla każdego kto posiada personalny komputer. „Świat nigdy już nie będzie ten sam" — twierdzi jeden ze współtwórców superkomputerów. „Wkrótce każda firma, każdy naukowiec, a nawet każdy krok życia będzie w pewien sposób związany z superkomputerami".

Szybkość — podstawowa miara mocy — stanowi główną różnicę między superkomputerami a ich mniejszymi krewniakami. Szybkość początkowo była mierzona tysiącami FLOPS'ów (akronim od „floating-point operations per second"), innymi słowy — liczb zmiennoprzecinkowych. Owa miara w dzisiejszych maszynach wynosi giga-FLOPS'y lub inaczej — superkomputery wykonują miliardy operacji na sekundę. Przewiduje się, że już wkrótce pojedyncza maszyna będzie pracowała z szybkością tera-FLOPS'ów, co daje moc dziesięciu milionów personalnych komputerów pracujących „na pełen gaz".

Superkomputery o największej mocy są zaskakująco małe i estetyczne (lśniące). Wewnątrz bardzo upakowane — by nie tracić czasu na przesyłanie informacji zbyt długimi drutami. Jak już wspomniano, maszyna ta może być połączona kablami lub via satelita z setkami terminali, które zamieniają szeregi cyfr na przykład na piękną kolorową trzywymiarową grafikę. Sprzęt ten, jak każde intensywnie pracujące urządzenie, musi być chłodzony. Bardzo sprytnie poradzono sobie z tym problemem w Uniwersytecie Minnesota — uzyskane tą drogą ciepło wykorzystywane jest do... ogrzewania garaży.

Rynek superkomputerów zdominowany został przez jedną firmę: CRAY Research of Minneapolis, która wyprodukowała 178 komputerów o charakterystycznym kształcie litery C (jak Ciechocinek). Owe superkomputery zainstalowane są na całym świecie i stanowią około 60% potencjału tego typu sprzętu. Najpoważniejszym konkurentem CRAY'a jest Control Data Corp. CDC, firma, która w 1983 roku stworzyła superkomputer ETA. Maszyny te, to około 13% światowych superkomputerów. Natomiast 23% tego typu komputerów wyprodukowano w Kraju Kwitnącej Wiśni przez NEC'a, Hitachi i Fujitsu, począwszy od 1983 roku.

Niespodziankę (oczekiwaną chyba przez wszystkich) szykuje gigant komputerowy IBM (sprzedaż w 1987 roku 54,2 miliarda dolarów), który zatrudnił (podkupił?) mistrza od superkomputerów Steve'a Chen'a („odzysk" z CRAY'a). Propozycja IBM'a, to komputer z 64 pracującymi równolegle procesorami — źródło znacznego przyspieszenia pracy komputera. Uznawane jest to jako znacząca nowość.

Okazuje się jednak, że rozwiązanie takie nie jest oryginalnym pomysłem IBM'a, mówi się bowiem, że inni projektanci tworzą maszyny z setkami, a nawet tysiącami procesorów. Wiosną 1988 roku naukowcy z Snandia National Labolatories w Albuquerque ogłosili, że ich 1024-procesorowy komputer jest 1000 razy szybszy od każdego komputera z pojedynczym procesorem, sugerując jednocześnie, iż opracowali metodę proporcjonalnego zwiększania szybkości komputera w zależności od liczby zainstalowanych w nim procesorów.

Większość badań nad superkomputerami w USA finansowana jest przez władze. Oczywiście najbardziej zainteresowane wzrostem mocy komputerów jest wojsko i wywiad. W 1987 roku Pentagon poświęcił setki milionów dolarów na doskonalenie superkomputerów. Ponadto sprzęt ten stanowi znaczne uwarunkowanie postępów w podboju przestrzeni kosmicznej. Jednym ze spektakularnych tego typu zastosowań jest wspomaganie prac projektowych nad samolotem „fruwającym" z szybkością 25 razy więk-

szą od głosu. Potrzeba taka wynikła z braku dostępu do tunelu aerodynamicznego, w którym można by było symulować eksploatację takiego wehikułu. Tradycyjny sposób testowania samolotów zastąpiony został w tym wypadku przez wykorzystanie komputera. Ponad 1,7 miliarda dolarów ma kosztować superkomputer w 1990 roku, który przekroczy szybkość mierzoną tera-FLOPS'ami.

Już od 1976 roku pierwszy CRAY pracował nad bombą wodorową. Inny „wczesny" CRAY przeznaczono do prac w National Security Agency, gdzie do dzisiaj rozwiązuje się „wojskowe zagadki", przegląda wywiadowcze dane, które każdego dnia napływają do agencji.

A inne zastosowania? Jest ich bardzo dużo. Na przykład przy projektowaniu sprzętu elektronicznego, finansiści radzą się superkomputerów, gdzie można opłacalnie inwestować, biomedycy zaprzęgają maszyny do analizy molekuł, by stwierdzić, które z nich mogą się stać nowymi lekami. Natomiast inżynierom CRAY pomaga projektować samochody, silniki, żaglówki, lodówki i nowe rośliny. Jednak najwięcej czasu z superkomputerami spędzają ludzie nauki. Stało się to osiągalne dzięki wspomnianym wcześniej połączeniom (via telefon). Na przykład superkomputer Narodowej Fundacji Nauki (koszt 200 mln dol.) „obsługuje" 6000 naukowców oraz 200 innych instytucji. Wspomniane przedsięwzięcia najczęściej wspomagane są przez różne wersje CRAY'a (CRAY-2, CRAY X-MP).

Mimo ogromnego postępu, użytkownicy superkomputerów nie są w pełni usatysfakcjonowani parametrami tego sprzętu. Potrzebne są superkomputery 100 razy szybsze. Wyścig w budowaniu szybszych superkomputerów jest na dobrej drodze. W dziesiątkach laboratoriów USA, Europy i Japonii prowadzone są bardzo intensywne prace nad doskonaleniem superkomputerów. Podstawowym celem jest zbudowanie najszybszej maszyny.

Podobnie jak Edison żarówkę, Bell telefon — Seymour CRAY stworzył superkomputer. Ten genialny inżynier zaprojektował: CDC 1604 (1960 rok), CDC (1964), CDC 7600 (1969), CRAY-1 (1976) i CRAY-2 (1985). Każdy z tych komputerów zawsze był uznawany w danym okresie za najlepszy. Obecnie ten wybitny konstruktor zajmuje się budową komputera CRAY-3, który ma być gotowy w 1988 roku i zaraz potem należy spodziewać się kolejnej maszyny — CRAY-4. CRAY-3 będzie pierwszym komercyjnym komputerem, który zbudowano korzystając z układów scalonych produkowanych na bazie arsenku galu (zamiast krzemu). Podłoże to jest 10-krotnie szybsze dla wędrujących elektronów. W czwartym modelu CRAY'a zostaną zainstalowane 64 procesory (numer 2 miał tylko cztery, w CRAY-3 zainstalowano 16 procesorów).

Zapewne usłyszymy jeszcze nieraz o wybitnym uczniu Cray'a 44-letnim Chińczyku Chen. Udoskonalił on konstrukcję swego mistrza i w efekcie z CRAY-1 powstał CRAY X-MP — dwuprocesorowa maszyna, jak dotychczas jest ona najlepiej sprzedawana (aż 120 instalacji). Kolejny wytwór Chena'a, to CRAY Y-MP. Jednak różnica pokoleń, nawet tam za oceanem, dała znać o sobie i Chen, którego praca (zdaniem Cray'a) charakteryzowała się zbyt dużym rozmachem musiał pożegnać się z nauczycielem. Wizjonerstwo Chen'a nie przestraszyło IBM'a, gdzie może realizować swoje plany, a sięgają one 100 miliardów giga-FLOPS'ów!

IBM poza projektem Chen'a ma „na warsztacie" sześć innych przedsięwzięć. Jedno z nich to GF-11, który zajmuje około 50 m^2, inny komputer to RP-3, ale najbardziej ambitnym wytworem IBM'a jest TF-1 zawierający 33000 dużej szybkości procesorów. Szybkość tej maszyny jest 2000 razy większa od dzisiejszych superkomputerów. IBM nie obawia się Cray'a, ale... konkurencji japońskiej — Hitachi, Fujitsu, i NEC'a. Superkomputery z Japonii (pierwsza generacja) świadczą o tym, iż amerykański przemysł komputerowy zostanie „zniszczony" w ciągu 25 lat. Współczesne maszyny made in Japan są porównywalne z najszybszymi made in USA, a w pewnych zastosowaniach są nawet lepsze. Porównując najnowszy pojedynczy procesor Hitachi S-820/80 z dwuprocesorowym Cray'a X-MP, Hitachi jest 10 razy lepszy. Japończycy skupili się na szybkich układach procesorowych. Przed nimi są jeszcze ogromne rezerwy w budowie komputerów z wieloma procesorami.

W komputerach równoległych (gdzie pracuje wiele procesorów jednocześnie), głównym problemem okazuje się oprogramowanie, całkowicie różne od oprogramowania komputerów jednoprocesorowych. To nowe oprogramowanie zaczyna dopiero „raczkować", mimo to efekty są widoczne — komputery równoległe są bezkonkurencyjne. Oprogramowanie na równi ze sprzętem decyduje o szybkości superkomputerów (równoległych). Powstały zatem warunki, wręcz konieczność połączenia wysiłków twórców techniki i specjalistów od oprogramowania, którzy dotychczas zajmowali się sztuczną inteligencją. Efekt owej współpracy najlepiej widać na przykładzie, obecnie najdoskonalszego równoległego komputera — Connection Machine. Urządzenie to zawiera 65 536 procesorów. Nie mniej zadziwiające jest to, że maszynę tę zaprojektował 31-letni Daniel Hillis. Na zakończenie warto zacytować jego wypowiedź: *„Tak jak nie można zbudować samolotu bez wcześniej skonstruowanego wystarczająco mocnego silnika, nie można tworzyć sztucznej inteligencji bez szybkich komputerów"*. Warto zatem nieco więcej uwagi poświęcić owej sztucznej inteligencji, która odradza się ostatnio, ale o tym już innym razem.

WŁODZIMIERZ GOGOŁEK

Opracowano na podstawie: TIME, March 28, 1988

SKOJARZENIE CZTERECH PAR

DANUTA KWASIŻUR, MIECZYSŁAW SKONIECZNY

Z szóstego rozdziału „LILAVATI" S. Jeleńskiego zatytułowanego „Z tajników szachownicy, kart i domina" wybraliśmy grę nr 2, którą chcemy zaprezentować Czytelnikom w dzisiejszym odcinku mikrokomputerowych anegdot matematycznych.

Zasady gry pojawiają się po uruchomieniu programu.

```
Na planszy gry zostaly rozlozone w rzad
cztery karty czerwonej i cztery czarnej
masci tak, ze barwy sie przeplataja:
czerwona-czarna-czerwona-czarna itd.
Po lewej stronie szeregu sa dwa miejsca
wolne. Na wolne miejsca mozna przeniesc
tylko dwie karty obok siebie lezace, nie
zmieniajac porzadku w jakim lezaly;i tak
samo na miejsce oproznione mozna znowu
przeniesc dwie karty sasiadujace.
Zadanie polega na tym, zeby w kilku
poruszeniach przesuwajac karty parami
ulozyc je w ten sposob, by lezaly kolo
siebie pod rzad cztery czarne, a za nimi
cztery czerwone karty. /Karty musza byc
ulozone parami/. Jezeli w czasie gry
zrezygnujesz z rozwiazywania tego wa-
tu wprowadz zamiast numeru karty '11'.
```

Po naciśnięciu dowolnego klawisza ukazuje się plansza gry, na której karty ułożone są w sposób losowy. Program oczekuje wprowadzenia dwóch numerów kart, które należy przesunąć. Muszą one leżeć obok siebie.

Sytuację po pierwszym przesunięciu przedstawia poniższy rys.


```
1 PRZESUNIECIE
PODAJ NUMER PIERWSZEJ KARTY 5
PODAJ NUMER DRUGIEJ KARTY   6
```

Gra prowadzona jest dalej w ten sam sposób do momentu poprawnego ułożenia kart. Wówczas pojawi się komunikat:

„KARTY PRAWIDŁOWO UŁOŻONE
PO n PRZESUNIĘCIACH
SPRÓBUJESZ JESZCZE RAZ? (T/N)"

i użytkownik może rozpocząć grę od początku z tym samym ułożeniem początkowym kart lub nowym, losowo wybranym przez program.

Wersja źródłowa programu jest następująca:

W segmencie sterującym wykorzystuje się 4 podprogramy, które realizują następujące funkcje:

1. Podprogramy z komunikatami o błędach: (instrukcje 900—960)
2. Podprogram badania sukcesu gry: (instrukcje 970—1200)
3. Podprogram rysowania dowolnej karty na ekranie: (instrukcje 1270—1410)
4. Podprogram losowego ustalenia początkowego ułożenia kart: (instrukcje 1420—1610)

```
10 '
20 ' Ekran informacyjny
30 '
40 MODE 1:INK 0,26:INK 1,0:INK 2,6:INK 3,9:PAPER 0:BORDER 26:CLS:PEN 1
50 SYMBOL AFTER 160
60 SYMBOL 160,3,15,31,31,63,63,63,63
70 SYMBOL 161,192,240,248,248,252,252,252,252
80 SYMBOL 162,63,63,63,63,31,31,15,3
90 SYMBOL 163,252,252,252,252,248,248,240,192
100 SYMBOL 164,63,63,63,63,63,63,63,63
110 SYMBOL 165,252,252,252,252,252,252,252,252
120 LOCATE 1,3
130 PRINT "Na planszy gry zostaly rozlozone w rzad"
140 PRINT "cztery karty czerwonej i cztery czarnej"
150 PRINT "masci tak, ze barwy sie przeplataja:  "
160 PRINT "czerwona-czarna-czerwona-czarna itd.  "
170 PRINT "Po lewej stronie szeregu sa dwa miejsca"
180 PRINT "wolne. Na wolne miejsca mozna przeniesc"
190 PRINT "tylko dwie karty obok siebie lezace, nie";
200 PRINT "zmieniajac porzadku w jakim lezaly;i tak";
210 PRINT "samo na miejsce oproznione mozna znowu "
220 PRINT "przeniesc dwie karty sasiadujace.     "
230 PRINT "Zadanie polega na tym, zeby w kilku   "
240 PRINT "poruszeniach przesuwajac karty parami  "
250 PRINT "ulozyc je w ten sposob, by lezaly kolo "
260 PRINT "siebie pod rzad cztery czarne, a za nimi";
270 PRINT "cztery czerwone karty. /Karty musza byc"
280 PRINT "ulozone parami/. Jezeli w czasie gry   "
290 PRINT "zrezygnujesz z rozwiazywania tego wa-  "
300 PRINT "tu wprowadz zamiast numery karty 11.   "
310 LOCATE 8,25:PRINT "NACISNIJ DOWOLNY KLAWISZ"
320 CLEAR INPUT
330 A$=INKEY$
340 IF A$="" THEN GOTO 330
350 '
360 ' Segment organizacyjny
370 '
380 MODE 1
390 DIM KARTY(10,2):DIM KARTY1(10,2)
400 GOSUB 1450
410 WINDOW £1,1,40,2,10:PAPER £1,1:CLS £1
420 NUM=1:NUM1=2
430 B$=CHR$(143)+CHR$(143):B1$=CHR$(164)+B$+CHR$(165)
440 FIG$(1)="D":FIG$(2)="K"
450 PAPER 1:PEN 0
460 FOR I=1 TO 10
470 LOCATE 2+(I-1)*4,3:PRINT USING "££";I
480 NEXT I
490 FOR K=1 TO 10
500 GOSUB 1300
510 NEXT K
520 LP=1
530 WINDOW £2,4,35,14,15:PAPER £2,3:PEN £2,0
540 WINDOW £3,4,35,16,18:PAPER £3,3:PEN £3,0
550 WINDOW £4,4,35,21,23:PAPER £4,2:PEN £4,0
560 WINDOW £5,4,35,12,13:PAPER £5,3:PEN £5,0:CLS £5
570 LOCATE £5,10,2:PRINT £5,"PRZESUNIECIE"
580 CLS £2:CLS £3
590 LOCATE £5,7,2:PRINT £5,USING "££";LP
600 CLS £2:LOCATE £2,2,2:INPUT £2,"PODAJ NUMER PIERWSZEJ KARTY ",NR
610 IF nr<1 OR nr>11 THEN GOTO 600
620 IF NR=11 THEN GOTO 1150
630 IF KARTY(NR,1)=0 THEN GOSUB 930:GOTO 600
640 IF nr=1 AND karty(2,1)=0 THEN GOSUB 950:GOTO 600
650 IF nr=10 AND karty(9,1)=0 THEN GOSUB 950:GOTO 600
660 IF karty(nr-1,1)=0 AND karty(nr+1,1)=0 THEN GOSUB 950:GOTO 600
```

```
670 CLS £3:LOCATE £3,2,2:INPUT £3,"PODAJ NUMER DRUGIEJ KARTY   ",NR1
680 IF nr1<1 OR nr1>11 THEN GOTO 670
690 IF NR1=11 THEN GOTO 1150
700 IF KARTY(NR1,1)=0 THEN GOSUB 930:GOTO 670
710 IF NR=NR1+1 OR NR=NR1-1 THEN GOTO 730
720 GOSUB 940:GOTO 670
730 IF NR<NR1 THEN GOTO 750
740 B=NR:NR=NR1:NR1=B
750 FOR L=1 TO 5
760 PEN 0:PAPER 1:LOCATE 1+(NR-1)*4,4+L:PRINT SPC(8);
770 NEXT L
780 KARTY(NUM,1)=KARTY(NR,1)
790 KARTY(NUM,2)=KARTY(NR,2)
800 KARTY(NUM1,1)=KARTY(NR1,1)
810 KARTY(NUM1,2)=KARTY(NR1,2)
820 KARTY(NR,1)=0:KARTY(NR,2)=0
830 KARTY(NR1,1)=0:KARTY(NR1,2)=0
840 K=NUM:GOSUB 1310
850 K=NUM1:GOSUB 1310
860 NUM=NR:NUM1=NR1
870 GOSUB 1020
880 LP=LP+1
890 GOTO 580

900 '
910 ' Podprogramy z komunikatami o bledach
920 '
930 PAPER £4,2:CLS £4:LOCATE £4,2,2:PRINT £4,"W TYM MIEJSCU NIE MA KARTY.":GOTO
960
940 PAPER £4,2:CLS £4:LOCATE £4,2,2:PRINT £4,"KARTY NIE LEZA OBOK SIEBIE.":GOTO
960
950 PAPER £4,2:CLS £4:LOCATE £4,2,2:PRINT £4,"PRZESUNIECIE NIEMOZLIWE":GOTO 960
960 SOUND 1,300:FOR T=1 TO 2000:NEXT T:PAPER £4,0:CLS £4:RETURN

970 '
980 ' Podprogram badania sukcesu gry
990 '
1000 IF karty(9,1)=0 AND karty(10,2)=0 THEN GOTO 1020
1010 RETURN
1020 FOR J=1 TO 10 STEP 2
1030 IF KARTY(J,2)=KARTY(J+1,2) THEN GOTO 1050
1040 RETURN
1050 NEXT J
1060 FOR J=1 TO 4
1070 IF KARTY(J,2)=2 OR KARTY(J,2)=3 THEN RETURN
1080 NEXT j
1090 CLS £5:CLS £2:CLS £3
1100 IF LP>10 THEN GOTO 1140
1110 LOCATE £5,12,2:PRINT £5,"BRAWO!!!"
1120 LOCATE £2,5,2:PRINT £2,"KARTY PRAWIDLOWO ULOZONE":PRINT £3,"    PO ";USING
"££";lp;:PRINT £3," PRZESUNIECIACH."
1130 GOTO 1150
1140 LOCATE £2,5,2:PRINT £2,"KARTY PRAWIDLOWO ULOZONE":PRINT £3,"    PO ";USING
"££";LP;:PRINT £3," PRZESUNIECIACH."
1150 PAPER £4,2:CLS £4:LOCATE £4,2,2:INPUT £4,"SPROBUJESZ JESZCZE RAZ(T/N)";T$
1160 IF T$="T" OR T$="t" THEN GOTO 1190
1170 IF T$="n" OR T$="N" THEN END
1180 GOTO 1150
1190 CLS £4:LOCATE £4,2,2:INPUT £4,"TEN SAM UKLAD KART (T/N)";T$
1200 IF T$="T" OR T$="t" THEN MODE 1:GOTO 1230
1210 IF T$="n" OR T$="N" THEN MODE 1:GOSUB 1430:GOTO 410
1220 GOTO 1190
1230 FOR K=1 TO 10
1240 KARTY(K,1)=KARTY1(K,1):KARTY(K,2)=KARTY1(K,2)
1250 NEXT K
1260 GOTO 410

1270 '
1280 ' Podprogram rysowania dowolnej karty
1290 '
1300 IF KARTY(K,1)=0 THEN RETURN
1310 PAPER 1:PEN 0
1320 LOCATE 1+(K-1)*4,5:PRINT CHR$(160);B$;CHR$(161);
1330 FOR I=6 TO 8
1340 LOCATE 1+(K-1)*4,I:PRINT B1$;
1350 NEXT I
1360 LOCATE 1+(K-1)*4,9:PRINT CHR$(162);B$;CHR$(163);
1370 PAPER 0
1380 IF KARTY(K,2)=1 OR KARTY(K,2)=4 THEN PEN 1:GOTO 1400
1390 PEN 2
1400 LOCATE 2+(K-1)*4,7:PRINT FIG$(KARTY(K,1));CHR$(225+KARTY(K,2))
1410 RETURN

1420 '
1430 ' Podprogram losowego ustalania poczatkowego ustawienia kart
1440 '
1450 KARTY(1,1)=0:KARTY(1,2)=0
1460 KARTY(2,1)=0:KARTY(2,2)=0
1470 karty(3,1)=1+INT(RND*2)
1480 karty(3,2)=1+INT(RND*4)
1490 FOR k=4 TO 10
1500 fig=1+INT(RND*2)
1510 kol=1+INT(RND*4)
1520 FOR i=3 TO k-1
1530 IF fig=karty(i,1) AND kol=karty(i,2) THEN GOTO 1500
1540 NEXT i
1550 karty(k,1)=fig
1560 karty(k,2)=kol
1570 NEXT k
1580 FOR K=1 TO 10
1590 KARTY1(K,1)=KARTY(K,1):KARTY1(K,2)=KARTY(K,2)
1600 NEXT K
1610 RETURN
```

KOMPUTEROWA ANALIZA UKŁADÓW ELEKTRONICZNYCH

ANALIZA STAŁOPRĄDOWA — DC

Chciałbym przybliżyć Czytelnikom metody wykorzystywane w komputerowej analizie układów elektronicznych. Ponieważ istnieje wiele profesjonalnych programów tego typu, nie silę się na stworzenie jeszcze jednej ich wersji. Postaram się w możliwie czytelny i ogólny sposób przedstawić algorytm działania tych programów.

Użytkownik wprowadza do komputera strukturę układu, parametry elementów elektronicznych (ewentualnie temperaturę otoczenia, tolerancje itp.). Na podstawie tych danych program generuje układ równań (w ogólnym wypadku są to nieliniowe równania różnicowe). Rezultatem obliczeń są najczęściej napięcia i prądy występujące w układzie.

Zagadnienie podzieliłem na kilka części tematycznych. W pierwszym odcinku proponuję zająć się **analizą stałoprądową** zwaną inaczej **analizą DC**.

Rozważmy prosty układ elektroniczny przedstawiony na **rys. 1**. Przypominam, że **źródło prądowe** wymusza przepływ stałego prądu między swoimi węzłami. **Węzłem** nazywamy punkt połączenia elementów (na schemacie punkty te oznaczone są kropkami i ponumerowane). Węzeł o numerze 0 *(zero)* jest węzłem odniesienia i stanowi tzw. **masę** układu elektronicznego. Względem niego są obliczane napięcia występujące na pozostałych węzłach.

Proponuję obliczyć napięcia występujące w układzie (oczywiście względem węzła 0) wykorzystując **I** i **II prawo Kirchhoffa**. Na schemacie zaznaczyłem rozpływ wszystkich prądów. Zgodnie z II prawem Kirchhoffa **suma prądów wpływających i wypływających z węzła wynosi zero**. Umówmy się: *prąd wypływający z węzła jest dodatni, a prąd do niego wpływający jest ujemny.* Analizując poszczególne węzły otrzymujemy:

węzeł 1: $I_1 + I_2 - J = 0$
węzeł 2: $-I_1 + I_3 - I_4 = 0$
węzeł 3: $-I_2 + I_0 - I_5 = 0$

Uwaga: Ponieważ suma prądów płynących w układzie wynosi 0 *(zero)*, to analiza węzła odniesienia nie ma sensu, gdy prądy wpływające i wypływające z tego węzła są w sposób jednoznaczny określone przez rozpływ prądów w pozostałych węzłach.

Chcemy, aby wynikiem naszych obliczeń były napięcia, stąd podstawmy zamiast prądów wyrażenia zawierające napięcia. Np. prąd i_1 na podstawie **prawa Ohma** można zapisać następująco:

$I_1 = \frac{U_{1-2}}{R_1}$, gdzie U_{1-2} jest napięciem występującym pomiędzy węzłami 1 i 2.

Ponieważ interesują nas napięcia względem masy stąd:

$$U_{1-2} = U_1 - U_2$$

Na tej podstawie prąd i_1 możemy zapisać jako:

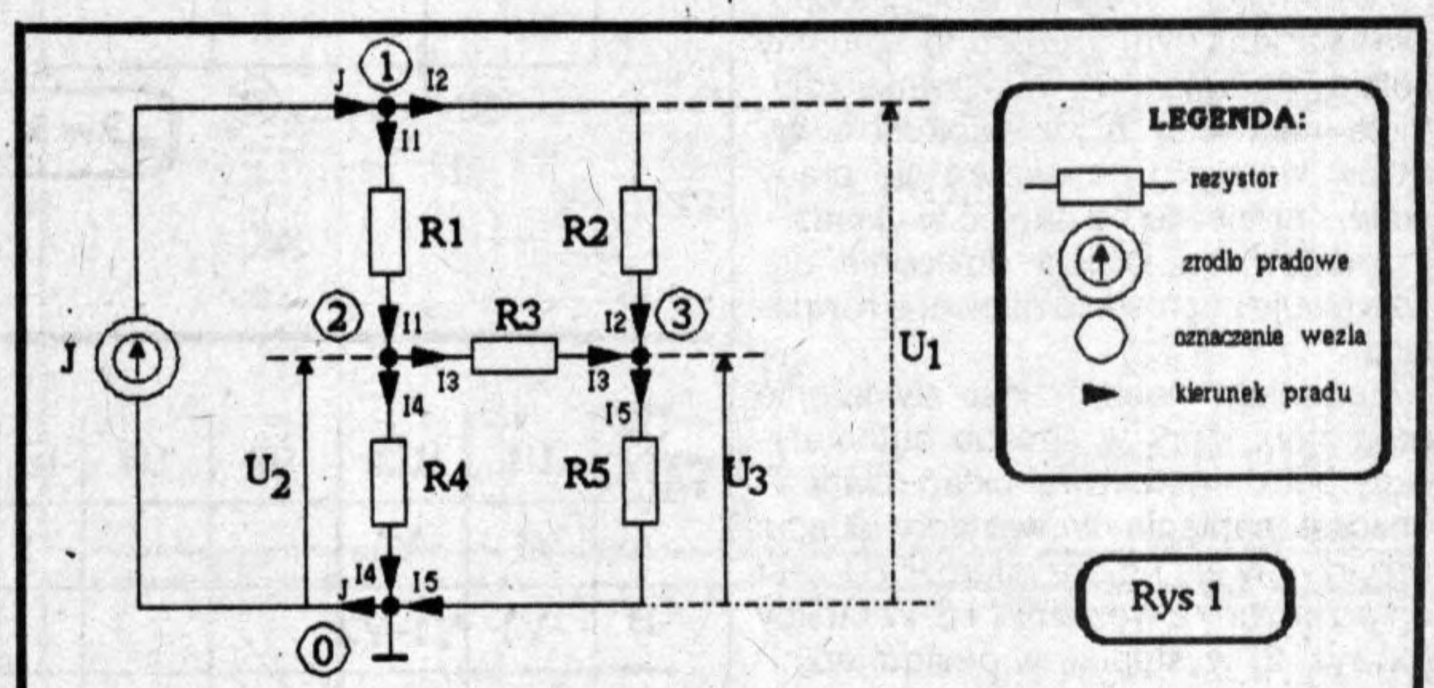
$$I_1 = \frac{U_1 - U_2}{R_1} = (U_1 - U_2) * Y_1,$$

gdzie

$Y_1 = \frac{1}{R_1}$ — **admitancja** (stosujemy admitancję aby uprościć zapis)

Analogicznie: $I_2 = (U_1 - U_3) * Y_2$
$I_3 = (U_2 - U_3) * Y_3$
$I_4 = U_2 * Y_4$
$I_5 = U_3 * Y_5$

Wyliczone prądy należy podstawić do bilansu prądów w węzłach. Po wymnożeniu i uporządkowaniu wyrażeń (proponuję czytelnikom samodzielnie wykonać te przekształcenia) otrzymu-

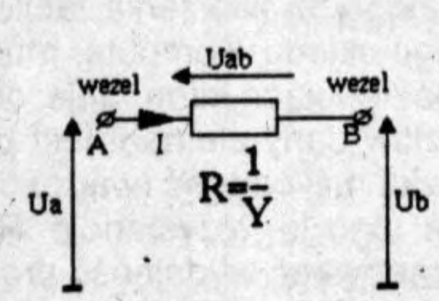


Rys 1

napięcie / węzeł	U1	U2	U3	b
1	Y1+**Y2**	-Y1+Y3	-**Y2**	J
2	-Y1	Y1+Y3+Y4	-Y3	0
3	-**Y2**	-Y3	**Y2**+Y3+Y5	0

UKLAD ROWNAN
dla schematu z rysunku 1
U1 , U2 , U3 - zmienne niewiadome
b - kolumna wyrazow stalych
1 , 2 , 3 - numery wezlow

Rys 2



napięcie / węzeł		Ua		Ub
A		**Y**		-Y
B		-Y		Y

Szablon dla rezystora
Uab-napiecie na rezystorze
Ua-napiecie na wezle A wzgl. masy
Ub-napiecie na wezle B wzgl. masy

Rys 3

jemy następujący układ równań:
węzeł 1: $(Y_1 + Y_2) * \mathbf{U_1} + (-Y_1 + Y_3) * * U_2 + (-Y_2) * \mathbf{U_3} = J$
węzeł 2: $(-Y_1) * U_2 + (Y_1 + Y_3 + Y_4) * \mathbf{U_2} + (-Y_3) * \mathbf{U_3} = 0$
$* U_2 + (-Y_2) * \mathbf{U_3} = J$
węzeł 3: $(-Y_2) * + U_1 + (-Y_3) * \mathbf{U_2} + + (Y_2 + Y_3 + Y_5) * \mathbf{U_3} = 0$

Otrzymaliśmy w ten sposób układ trzech równań z trzema niewiadomymi: $\mathbf{U_1}, \mathbf{U_2}, \mathbf{U_3}$. Prąd źródła stałego (ponieważ nie zależy od żadnego napięcia) został przeniesiony na prawą stronę równania jako wartość stała.

Osiągnęliśmy pierwszy sukces: wiemy jak na podstawie schematu, z praw Kirchhoffa, stworzyć układ równań matematycznych obliczających napięcia we wszystkich węzłach układów względem węzła odniesienia (masy).

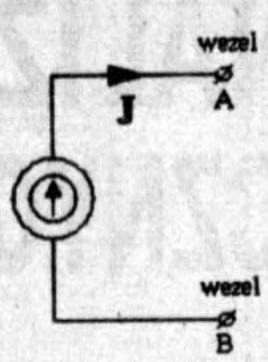
Na rys. 2 przedstawiłem tablicowy sposób zapisu tego równania. Będzie on bardzo pomocny w późniejszych rozważaniach. Wiersze tablicy odpowiadają kolejnym węzłom, a kolumny kolejnym napięciom. W ostatniej kolumnie (nazwanej **„b"**, tzw. kolumnie wyrazów wolnych) umieszcza się prądy stałe, niezależne od napięć w układzie. Przedstawiona tablica dokładnie odwzorowuje poprzednio napisane równania.

Zadaniem naszym jest stworzenie programu, który w sposób automatyczny będzie generował układ równań i obliczał napięcia w węzłach układu. Przyjrzyjmy się np. rezystorowi R_2. Jest on połączony z węzłami 1 i 3. W tablicy (na rys. 2) występuje w postaci admitancji Y_2. Zaznaczyłem ją dla wyróżnienia grubszą czcionką. Okazuje się, że występowanie admitancji Y_2 w tablicy jest określone przez węzły, do których ten rezystor jest dołączony: znajdują się na przecięciu wierszy 1 i 3 oraz kolumn U_1 i U_3. Na rys. 3 przedstawiłem uogólniony rezystor o rezystancji **R** (admitancji Y) podłączony do węzłów **A** i **B.** Uproszczona tablica obok przedstawia miejsca występowania admitancji Y w rzeczywistej tablicy układu. Niedowiarkom proponuję proste sprawdzenie: rozważmy występowanie admitancji dla naszego rezystora R_2: wtedy węzłowi A będzie odpowiadać węzeł 1, węzłowi B węzeł 3, napięciu Ua napięcie U_1, a napięciu Ub napięcie U_2. Okazuje się, że admitancje Y pokrywają się z admitancjami Y_2. Proponuję czytelnikom sprawdzenie zgodności tej metody dla rezystorów R_1 i R_3. Uproszczona tablica z rys. 3 nazywana jest **szablonem.** W omawianym wypadku jest to szablon dla rezystora. Każdy element występujący w układzie posiada swój szablon. Wtedy tworzenie tablicy dla dowolnego układu staje się bardzo proste: zgodnie z szablonem danego elementu dołączamy do tablicy układu odpowiednie współczynniki tego elementu (dla rezystora będzie to jego admitancja). Znaleźliśmy w ten sposób algorytm tworzenia tablicy dla dowolnego układu. Komputer musi posiadać następujące informacje: do których węzłów dany element jest podłączony i jaką ma wartość (wartością dla rezystora będzie rezystancja R, dla źródła prądowego wydajność prądowa J itp.) Na podstawie szablonu danego elementu i wyżej wymienionych informacji do odpowiednich miejsc tablicy układu dołączy współczynniki tego elementu.

Dociekliwi czytelnicy na pewno spostrzegli rażące odstępstwo od tych zasad dla rezystorów R_4 i R_5. Jednak tak nie jest: elementy te podłączone są do węzła zerowego, czyli węzła masy; a wszystko to, co dotyczy tego węzła jest odrzucane. Przypatrzmy się jeszcze raz szablonowi z rys. 3. Jeżeli np. węzeł B będzie węzłem masy, to wiersz szablonu opisujący węzeł B i kolumnę napięcia Ub należy odrzucić — zostanie wtedy jeden współczynnik Y na przecięciu wiersza A i kolumny Ua (współczynnik ten zaznaczony jest dla wyróżnienia grubszą czcionką). Taka właśnie sytuacja zachodzi dla rezystorów R_4 i R_5.

Reguła ta dotyczy wszystkich elementów!

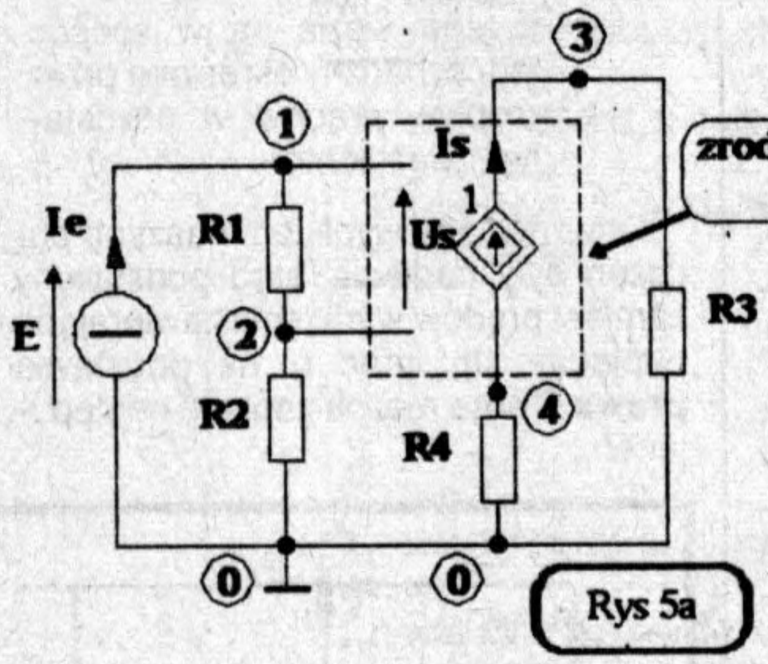
Podobna sytuacja zachodzi dla źród-



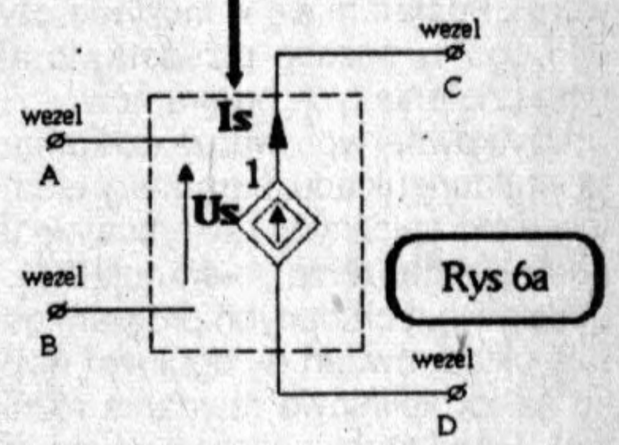
Rys 4

węzeł		b
A		J
B		-J

Szablon dla zrodła pradowego o wydajnosci pradowej J



Rys 5a



Rys 6a

napięcie / węzeł / gałąź	U1	U2	U3	U4	Ie	b
A	Y1	-Y1			-1	
B	-Y1	Y1+Y2				
C	-ys	ys	Y3			
D	ys	-ys		Y4		
gałąź źródła napiec.	-1					-E

Rys 5b

		Ua		Ub
C		-ys		ys
D		ys		-ys

Rys 6b

ła prądowego, szablon którego przedstawiłem na rys. 4.

Dużą grupę elementów stanowią tzw. **źródła sterowane.** Rozważmy prosty schemat (rys. 5a) zawierający *źródło prądowe sterowane napięciem.* Źródło takie wymusza prąd Ju zależny od napięcia Us. W najprostszej wersji, którą będziemy stosować, **Ju = ys*Us** (**ys** jest współczynnikiem podawanym przez użytkownika). Do zasilania układu zastosowałem źródło napięciowe o napięciu E. Wprowadzi ono trochę zamieszania, bowiem bilans prądów w węzłach będzie wymagał przyjęcia nowej zmiennej prądu źródła **Ie**:

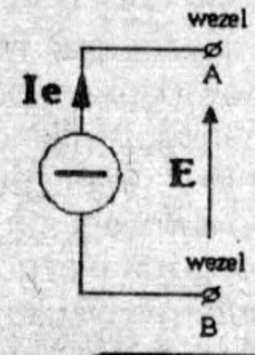
węzeł 1: $-Ie + I_1 = 0$
węzeł 2: $-I_1 + I_2 = 0$
węzeł 3: $-Is + I_3 = 0$
węzeł 4: $-I_4 + Is = 0$

równanie gałęziowe źródła E: $E = U_1$
gdzie: $I_1 = (U_1 - U_1) * Y_1$, $I_2 = U^2 * Y_2$, $I_3 = U_3 * Y_3$, $I_4 = U_4 * Y_4$, $Is = Us * ys = (U_1 - U_2) * ys$

Na rys. 5b przedstawiłem tablicę układu. Proponuję porównać z nią szablon modelu źródła prądowego (rys. 6a, rys. 6b). Na rys. 7 przedstawiłem szablon źródła napięciowego.

Uwaga: Wszystkie źródła o charakterze napięciowym wymagają wprowadzenia dodatkowej zmiennej, jaką jest prąd przez nie przepływający. Ta dodatkowa zmienna wymaga dołączenia jeszcze jednego równania, tzw. **równania gałęziowego** opisującego napięcie na węzłach, do których podłączone jest źródło.

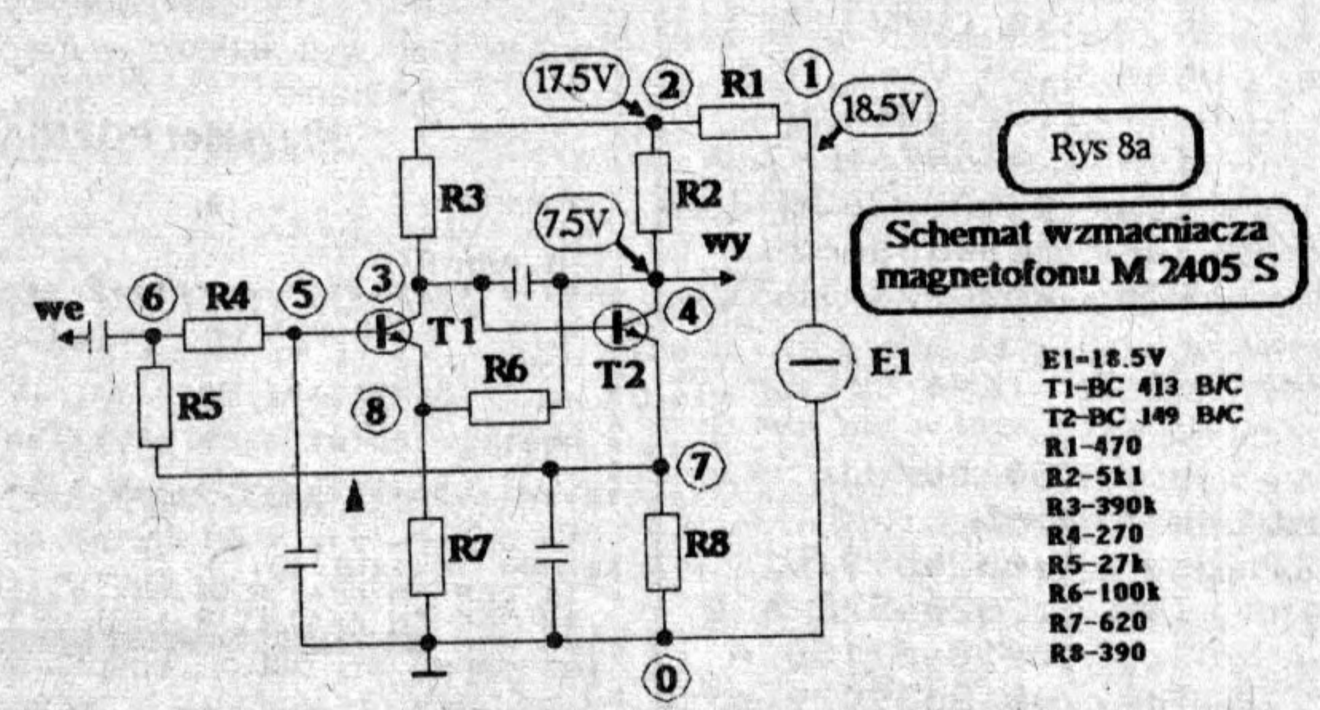
Na rys. 8a przedstawiłem schemat wzmacniacza wstępnego magnetofonu szpulowego M 2405 S. Ten układ poddałem stałoprądowej analizie numerycznej. Na rys. 8b przedstawiłem model tranzystora NPN, jaki stosuję w swoim programie: składa się on ze stałego źródła napięciowego i źródła prądowego sterowanego prądem przepływającym przez źródło napięciowe. Okazuje się, że nawet tak proste modele zapewniają dosyć dobrą dokładność.



Rys 7

Szablon stalego zrodla napieciowego

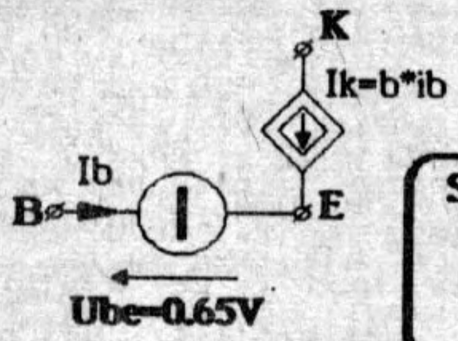
napięcie / węzeł / gałąź		Ua		Ub		Ie		b	
A						-1			
B						1			
gałąź zr.napiec.		-1		1				-E	



Rys 8a

Schemat wzmacniacza magnetofonu M 2405 S

E1-18.5V
T1-BC 413 B/C
T2-BC 149 B/C
R1-470
R2-5k1
R3-390k
R4-270
R5-27k
R6-100k
R7-620
R8-390



Rys 8b

Stałopradowy model tranzystora NPN w stanie aktywnym
b="beta"-stalopradowe wzmocnienie
Ube-napiecie: baza-emiter

```
napiecia w wezlach ukladu:
wezel 1 = 18.5 V
wezel 2 = 17.575459 V

wezel 3 = 1.3730188 V
wezel 4 = 7.7550787 V
wezel 5 = 0.72019338 V
wezel 6 = 0.72022136 V
wezel 7 = 0.72301877 V
wezel 8 = .070193382 V

SZN1   U+=1   U-=0

       U = 18.5 V
       I = .0019671094 A
       P = .036391524 W

R1     U+=1   U-=2

       R = 470 Om
       U = 0.92454142 V
       I = .0019671094 A
       P = .0018186741 W

R2     U+=2   U-=4

       R = 5100 Om
       U = 9.8203799 V
       I = .0019255647 A
       P = .018909777 W

R3     U+=2   U-=3

       R = 390000 Om
       U = 16.20244 V
       I = .000041544717 A
       P = .00067312578 W

R4     U+=6   U-=5

       R = 270 Om
       U = .000027973903 V
       I = 1.0360705E-7 A
       P = 2.8982936E-12 W

R5     U+=7   U-=6

       R = 27000 Om
       U = .002797409 V
       I = 1.0360774E-7 A
       P = 2.8983322E-10 W

R6     U+=4   U-=8

       R = 100000 Om
```

```
U = 7.6848853 V
I = .000076848853 A
P = .00059057462 W


R7      U+=8   U-=0

        R = 620 Om
        U = .070193382 V
        I = .00011321513 A
        P = 7.9469531E-6 W

R8      U+=7   U-=0

        R = 390 Om
        U = 0.72301877 V
        I = .0018538943 A
        P = .0013404003 W

NPN1    B=5    K=3    E=8

        b = 350 [V/V]
      Ube= 0.65 V
      Uce= 1.3028254 V
       Ib= 1.0360764E-7 A
       Ic= .000036262672 A
       Ie= .00003636628
        P = .000047311275 W

NPN2    B=3    K=4    E=7

        b = 350 [V/V]
      Ube= 0.65 V
      Uce= 7.0320599 V
       Ib= 5.2820452E-6 A
       Ic= .0018487158 A
       Ie= .0018539979
        P = .013003714 W
```

Program dopuszcza posługiwanie się 9 elementami:
SZP — stałe źródło prądowe
SZN — stałe źródło napięciowe
ZPSN — źródło prądowe sterowane napięciem
ZPSP — źródło prądowe sterowane prądem
ZNSN — źródło napięciowe sterowane napięciem
ZNSP — źródło napięciowe sterowane prądem
R — rezystor
NPN — tranzystor bipolarny NPN w stanie aktywnym
PNP — tranzystor bipolarny PNP w stanie aktywnym
Dodatkowo można zasymulować:
rozwarcie: źródło prądowe o zerowej wydajności prądowej
zwarcie: źródło napięciowe o napięciu E = 0
wzmacniacz operacyjny: źródło napięciowe sterowane napięciem o współczynniku i równym wzmocnieniu wzmacniacza operacyjnego

Użytkownik za pomocą edytora wprowadza strukturę układu do komputera: podaje węzły, do których elementy są podłączone i ich współczynniki. Program wykonuje analizę dwuetapowo:

etap 1: następuje stworzenie macierzy (odpowiednik naszej tablicy) danego układu. Jednocześnie następuje kontrola poprawności struktury układu. Każdą niezgodność komputer wypisuje na ekranie. Po stwierdzeniu błędu program nie przechodzi do drugiego etapu.

etap 2: komputer rozwiązuje układ równań na podstawie macierzy stworzonej w etapie I wykorzystując tzw. **metodę Gaussa** z wyborem elementu głównego. Metoda ta, choć niezbyt szybka, jest bardzo dokładna numerycznie. Efektem końcowym jest wyświetlenie wyników na ekranie.

Krzysztof POŹNIAK

```
  1 REM
**********************************
*                                *
*                    ©1988       *
*                                *
*      Krzysztof Poźniak         *
*                                *
* Analiza DC uklados elektron.   *
*                                *
**********************************

  10 DEF FN z(p,k,z)=(z<p) OR (z
>k) OR (z<>INT z)
  11 DEF FN u(k)=t(2,e(x,l,k))
  12 DEF FN p(k)=t(2,k+9)
 100 REM [illegible]
 105 PAPER 0: INK 7: BORDER 0: C
LS
 110 LET maxw=10: LET maxn=20
 120 GO SUB 7000
 130 GO SUB 7500
 140 GO TO 9000

1000 REM [illegible]
1010 LET n=lz-1: FOR k=1 TO n: L
ET x(k)=k: NEXT k: FOR k=1 TO n:
 LET m=0: FOR y=k TO n: FOR x=k
TO n: IF ABS n(y,x)>m THEN LET m
=ABS n(y,x): LET my=y: LET mx=x
1020 NEXT x: NEXT y: IF m=0 THEN
 PRINT '"uklad nie ma rozwiazani
a ": RETURN
1030 FOR l=1 TO n: LET m=n(k,l):
 LET n(k,l)=n(my,l): LET n(my,l)
=m: NEXT l: LET m=b(k): LET b(k)
=b(my): LET b(my)=m: FOR l=1 TO
n: LET m=n(l,k): LET n(l,k)=n(l,
mx): LET n(l,mx)=m: NEXT l: LET
m=x(k): LET x(k)=x(mx): LET x(mx
)=m
1040 FOR l=k+1 TO n: LET m=n(l,k
)/n(k,k): FOR j=k+1 TO n: LET n(
l,j)=n(l,j)-n(k,j)*m: NEXT j: LE
T b(l)=b(l)-b(k)*m: NEXT l: NEXT
 k: FOR k=n TO 1 STEP -1: LET m=
0: FOR l=k+1 TO n: LET m=m+t(1,x
(l))*n(k,l): NEXT l: LET t(1,x(k
))=(b(k)-m)/n(k,k): NEXT k: LET
m=1: PRINT '"analiza numeryczna
zakonczona": RETURN
5000 REM [illegible]
5001 REM uwagi i bledy
5005 IF e(x,l,2)=e(x,l,3) THEN P
RINT "uwaga-zwarte zrodlo SZP";e
(x,l,1)
5006 IF e(x,l,4)=0 THEN PRINT "u
waga-zerowa wyd. pradowa SZP";e(
x,l,1)
5010 REM tworzenie macierzy
5015 LET k=3: GO SUB 6100: IF st
op=1 THEN RETURN
5020 IF e(x,l,2)>0 THEN LET b(FN
 u(2))=b(FN u(2))+e(x,l,4)
5021 IF e(x,l,3)>0 THEN LET b(FN
 u(3))=b(FN u(3))-e(x,l,4)
5022 RETURN
5030 REM wizualizacja wynikow

5035 PRINT ' INVERSE 1;"SZP";e(x
,l,1); INVERSE 0;TAB 6;"U+=";e(x
,l,2);TAB 12;"U-=";e(x,l,3)
5036 LET m1=2: LET m2=3: GO SUB
6400: PRINT 'TAB 6;"U = ";m;" V"
;TAB 6;"I = ";e(x,l,4);" A": PRI
NT TAB 6;"P = ";m*e(x,l,4);" W"
5040 RETURN
5100 REM [illegible]
5105 IF e(x,l,4)=0 THEN PRINT "u
waga-zerowa wartosc nap. SZN";e(
x,l,1)
5106 IF e(x,l,2)=e(x,l,3) THEN P
RINT INVERSE 1;"blad-zwarte zrod
lo SZN";e(x,l,1): LET blad=1: RE
TURN
5110 REM tworzenie macierzy
5115 LET k=3: GO SUB 6100: LET k
=1: GO SUB 6200: IF stop=1 THEN
RETURN
5120 IF e(x,l,2)>0 THEN LET n(FN
 u(2),FN p(lp))=1: LET n(FN p(lp
),FN u(2))=-1
5121 IF e(x,l,3)>0 THEN LET n(FN
 u(3),FN p(lp))=-1: LET n(FN p(l
p),FN u(3))=1
5122 LET b(FN p(lp))=-e(x,l,4)
5125 RETURN
5130 REM wizualizacja wynikow
5135 PRINT ' INVERSE 1;"SZN";e(x
,l,1); INVERSE 0;TAB 6;"U+=";e(x
,l,2);TAB 12;"U-=";e(x,l,3)
5136 LET m1=2: LET m2=3: GO SUB
6400: PRINT 'TAB 6;"U = ";m;" V"
;TAB 6;"I = ";-t(1,FN p(lp));" A
": PRINT TAB 6;"P = ";-m*t(1,FN
p(lp));" W"
5140 LET lp=lp+1: RETURN
5200 REM [illegible]
5201 REM uwagi i bledy
5205 IF e(x,l,2)=e(x,l,3) THEN P
RINT "uwaga-zwarte wezly ster. Z
PSN";e(x,l,1)
5206 IF e(x,l,4)=e(x,l,5) THEN P
RINT "uwaga-zwarte wezly prad. Z
PSN";e(x,l,1)
5207 IF e(x,l,6)=0 THEN PRINT IN
VERSE 1;"Blad-zerowy wspolczynni
k ZPSN";e(x,l,1): LET stop=1: RE
TURN
5210 REM tworzenie macierzy
5215 LET k=5: GO SUB 6100: IF st
op=1 THEN RETURN
5220 IF e(x,l,2)>0 AND e(x,l,4)>
0 THEN LET n(FN u(4),FN u(2))=n(
FN u(4),FN u(2))-e(x,l,6)
5221 IF e(x,l,2)>0 AND e(x,l,5)>
0 THEN LET n(FN u(5),FN u(2))=n(
FN u(5),FN u(2))+e(x,l,6)
5222 IF e(x,l,3)>0 AND e(x,l,4)>
0 THEN LET n(FN u(4),FN u(3))=n(
FN u(4),FN u(3))+e(x,l,6)
5223 IF e(x,l,3)>0 AND e(x,l,5)>
0 THEN LET n(FN u(5),FN u(3))=n(
FN u(5),FN u(3))-e(x,l,6)
5225 RETURN
5230 REM wizualizacja wynikow
5235 PRINT ' INVERSE 1;"ZPSN";e(
x,l,1); INVERSE 0;TAB 6;"S+=";e(
x,l,2);TAB 12;"S-=";e(x,l,3);TAB
 18;"I+=";e(x,l,4);TAB 24;"I-=";
e(x,l,5)
5236 LET m1=2: LET m2=3: GO SUB
6400: LET u1=m: LET m1=4: LET m2
=5: GO SUB 6400: PRINT 'TAB 6;"U
r=";e(x,l,6);TAB 6;"Us= ";u1;" V
";TAB 6;"U = ";m;" V";TAB 6;"I =
";u1*e(x,l,6);" A": PRINT TAB 6
;"P = ";m*u1*e(x,l,6);" W"
5240 RETURN
5300 REM [illegible]
5301 REM uwagi i bledy
5305 IF e(x,l,2)=e(x,l,3) THEN P
RINT "uwaga-zwarte wezly ster. Z
PSP";e(x,l,1)
5306 IF e(x,l,4)=e(x,l,5) THEN P
RINT "uwaga-zwarte wezly prad. Z
PSP";e(x,l,1)
5307 IF e(x,l,6)=0 THEN PRINT IN
VERSE 1;"Blad-zerowy wspolczynni
k ZPSP";e(x,l,1): LET stop=1: RE
TURN
5310 REM tworzenie macierzy
5315 LET k=5: GO SUB 6100: LET k
=1: GO SUB 6200: IF stop=1 THEN
RETURN
5320 IF e(x,l,2)>0 THEN LET n(FN
 u(2),FN p(lp))=1: LET n(FN p(lp
),FN u(2))=-1
5321 IF e(x,l,3)>0 THEN LET n(FN
 u(3),FN p(lp))=-1: LET n(FN p(l
p),FN u(3))=1
5322 IF e(x,l,4)>0 THEN LET n(FN
 u(4),FN p(lp))=-e(x,l,6)
5323 IF e(x,l,5)>0 THEN LET n(FN
 u(5),FN p(lp))=e(x,l,6)
5325 RETURN
5330 REM wizualizacja wynikow
5335 PRINT ' INVERSE 1;"ZPSP";e(
x,l,1); INVERSE 0;TAB 6;"S+=";e(
x,l,2);TAB 12;"S-=";e(x,l,3);TAB
 18;"I+=";e(x,l,4);TAB 24;"I-=";
e(x,l,5)
5336 LET m1=4: LET m2=5: GO SUB
6400: PRINT 'TAB 6;"Uk=";e(x,l,6
);TAB 6;"Is= ";t(1,FN p(lp));" A
";TAB 6;"U = ";m;" V";TAB 6;"I =
";t(1,FN p(lp))*e(x,l,6);" A":
PRINT TAB 6;"P = ";m*t(1,FN p(lp
))*e(x,l,6);" W"
5340 LET lp=lp+1: RETURN
5400 REM [illegible]
5401 REM uwagi i bledy
5405 IF e(x,l,2)=e(x,l,3) THEN P
RINT "uwaga-zwarte wezly ster. Z
NSN";e(x,l,1)
5406 IF e(x,l,6)=0 THEN PRINT "U
waga-zerowy wspolcz. ZNSN";e(x,l
,1)
```

```
5407 IF e(x,l,4)=e(x,l,5) THEN P
RINT INVERSE 1;"blad-zwarte wezl
y nap. ZNSN";e(x,l,1): LET stop=
1: RETURN
5410 REM tworzenie macierzy
5415 LET k=5: GO SUB 6100: LET k
=1: GO SUB 6200: IF stop=1 THEN
RETURN
5420 IF e(x,l,2)>0 THEN LET n(FN
 p(lp),FN u(2))=-e(x,l,6)
5421 IF e(x,l,3)>0 THEN LET n(FN
 p(lp),FN u(3))=e(x,l,6)
5422 IF e(x,l,4)>0 THEN LET n(FN
 u(4),FN p(lp))=-1: LET n(FN p(l
p),FN u(4))=1
5423 IF e(x,l,5)>0 THEN LET n(FN
 u(5),FN p(lp))=1: LET n(FN p(lp
),FN u(5))=-1
5425 RETURN
5430 REM wizualizacja wynikow
5435 PRINT ' INVERSE 1;"ZPSP";e(
x,l,1); INVERSE 0;TAB 6;"S+=";e(
x,l,2);TAB 12;"S-=";e(x,l,3);TAB
 18;"I+=";e(x,l,4);TAB 24;"I-=";
e(x,l,5)
5436 LET m1=2: LET m2=3: GO SUB
6400: PRINT 'TAB 6;"Ur=";e(x,l,6
);TAB 6;"Us= ";m;" V";TAB 6;"U =
 ";m*e(x,l,6);" V";TAB 6;"I = ";
t(1,FN u(lp));" A": PRINT TAB 6;
"P = ";m*e(x,l,6)*t(1,FN u(lp));
" W"
5440 LET lp=lp+1: RETURN
5500 REM [illegible]
5501 REM uwagi i bledy
5505 IF e(x,l,2)=e(x,l,3) THEN P
RINT "uwaga-zwarte wezly ster. Z
N5P";e(x,l,1)
5506 IF e(x,l,6)=0 THEN PRINT "U
waga-zerowy wspolcz. ZN5P";e(x,l
,1)
5507 IF e(x,l,4)=e(x,l,5) THEN P
RINT INVERSE 1;"blad-zwarte wezl
y nap. ZN5P";e(x,l,1): LET stop=
1: RETURN
5510 REM tworzenie macierzy
5515 LET k=5: GO SUB 6100: LET k
=2: GO SUB 6200: IF stop=1 THEN
RETURN
5520 IF e(x,l,2)>0 THEN LET n(FN
 u(2),FN p(lp-1))=1: LET n(FN p(
lp-1),FN u(2))=-1
5521 IF e(x,l,3)>0 THEN LET n(FN
 u(3),FN p(lp-1))=-1: LET n(FN p
(lp-1),FN u(3))=1
5522 IF e(x,l,4)>0 THEN LET n(FN
 u(4),FN p(lp))=1: LET n(FN p(lp
),FN u(4))=-1
5523 IF e(x,l,5)>0 THEN LET n(FN
 u(5),FN p(lp))=-1: LET n(FN p(l
p),FN u(5))=1
5524 LET n(FN p(lp),FN p(lp-1))=
e(x,l,6)
5525 RETURN
5530 REM wizualizacja wynikow
5535 PRINT ' INVERSE 1;"ZN5P";e(
x,l,1); INVERSE 0;TAB 6;"S+=";e(
x,l,2);TAB 12;"S-=";e(x,l,3);TAB
 18;"U+=";e(x,l,4);TAB 24;"U-=";
e(x,l,5)
5536 PRINT 'TAB 6;"Ur=";e(x,l,6)
;TAB 6;"Is= ";t(1,FN p(lp));" A"
;TAB 6;"U = ";t(1,FN p(lp))*e(x,
l,6);" V";TAB 6;"I = ";t(1,FN p(
lp+1));" A": PRINT TAB 6;"P = ";
e(x,l,6)*t(1,FN p(lp))*t(1,FN p(
lp+1));" W"
5540 LET lp=lp+2: RETURN
5600 REM [illegible]
5601 REM uwagi i bledy
5605 IF e(x,l,2)=e(x,l,3) THEN P
RINT "uwaga-zwarty rezystor R";e
(x,l,1)
5606 IF e(x,l,4)=0 THEN PRINT IN
VERSE 1;"Blad-zerowa rezystancja
 R";e(x,l,1): LET stop=1: RETURN
5610 REM tworzenie macierzy
5615 LET k=3: GO SUB 6100
5620 IF e(x,l,2)>0 THEN LET n(FN
 u(2),FN u(2))=n(FN u(2),FN u(2)
)+1/e(x,l,4): IF e(x,l,3)>0 THEN
 LET n(FN u(2),FN u(3))=n(FN u(2
),FN u(3))-1/e(x,l,4)
5621 IF e(x,l,3)>0 THEN LET n(FN
 u(3),FN u(3))=n(FN u(3),FN u(3)
)+1/e(x,l,4): IF e(x,l,2)>0 THEN
 LET n(FN u(3),FN u(2))=n(FN u(3
),FN u(2))-1/e(x,l,4)
5625 RETURN
5630 REM wizualizacja wynikow
5635 PRINT ' INVERSE 1;"R";e(x,l
,1); INVERSE 0;TAB 6;"U+=";e(x,l
,2);TAB 12;"U-=";e(x,l,3)
5636 LET m1=2: LET m2=3: GO SUB
6400: PRINT 'TAB 6;"R = ";e(x,l,
4);" Om";TAB 6;"U = ";m;" V";TAB
 6;"I = ";m/e(x,l,4);" A": PRINT
 TAB 6;"P = ";m*m/e(x,l,4);" W"
5640 RETURN
5700 REM [illegible]
5701 REM uwagi i bledy
5705 IF e(x,l,2)=e(x,l,3) THEN P
RINT "uwaga-zwarte: B i K NPN";e
(x,l,1)
5706 IF e(x,l,3)=e(x,l,4) THEN P
RINT "uwaga-zwarte: K i E NPN";e
(x,l,1)
5707 IF e(x,l,5)=0 THEN PRINT "u
waga-zerowe wzmocnienie NPN";e(x
,l,1)
5708 IF e(x,l,2)=e(x,l,4) THEN P
RINT INVERSE 1;"blad-zwarte: B i
 E NPN";e(x,l,1): LET stop=1: RE
TURN
5710 REM tworzenie macierzy
5715 LET k=4: GO SUB 6100: LET k
=1: GO SUB 6200: IF stop=1 THEN
RETURN
5720 IF e(x,l,2)>0 THEN LET n(FN
 u(2),FN p(lp))=1: LET n(FN p(lp
),FN u(2))=-1
5721 IF e(x,l,3)>0 THEN LET n(FN
 u(3),FN p(lp))=e(x,k,5)
5722 IF e(x,l,4)>0 THEN LET n(FN
 u(4),FN p(lp))=-1-e(x,l,5): LET
 n(FN p(lp),FN u(4))=1
5723 LET b(FN p(lp))=-.65
5725 RETURN
5730 REM wizualizacja wynikow
5735 PRINT ' INVERSE 1;"NPN";e(x
,l,1); INVERSE 0;TAB 6;" B=";e(x
,l,2);TAB 12;" K=";e(x,l,3);TAB
18;" E=";e(x,l,4)
5736 LET m1=3: LET m2=4: GO SUB
6400: PRINT 'TAB 6;"b = ";e(x,l,
5);" [V/V]";TAB 5;"Ube= 0.65 V";
TAB 5;"Uce= ";m;" V";TAB 6;"Ib=
";t(1,FN p(lp));" A";TAB 6;"Ic=
";e(x,l,5)*t(1,FN p(lp));" A";TA
B 6;"Ie= ";t(1,FN p(lp))*(1+e(x,
l,5));TAB 6;"P = ";(.65+m*e(x,l,
5))*t(1,FN p(lp));" W"
5740 LET lp=lp+1: RETURN
5800 REM [illegible]
5801 REM uwagi i bledy
5805 IF e(x,l,2)=e(x,l,3) THEN P
RINT "uwaga-zwarte: B i K PNP";e
(x,l,1)
5806 IF e(x,l,3)=e(x,l,4) THEN P
RINT "uwaga-zwarte: K i E PNP";e
(x,l,1)
5807 IF e(x,l,5)=0 THEN PRINT "u
waga-zerowe wzmocnienie PNP";e(x
,l,1)
5808 IF e(x,l,2)=e(x,l,4) THEN P
RINT INVERSE 1;"blad-zwarte: B i
 E PNP";e(x,l,1): LET stop=1: RE
TURN
5810 REM tworzenie macierzy
5815 LET k=4: GO SUB 6100: LET k
=1: GO SUB 6200: IF stop=1 THEN
RETURN
5820 IF e(x,l,2)>0 THEN LET n(FN
 u(2),FN p(lp))=-1: LET n(FN p(l
p),FN u(2))=1
5821 IF e(x,l,3)>0 THEN LET n(FN
 u(3),FN p(lp))=-e(x,k,5)
5822 IF e(x,l,4)>0 THEN LET n(FN
 u(4),FN p(lp))=1+e(x,l,5): LET
n(FN p(lp),FN u(4))=-1
5823 LET b(FN p(lp))=-.65
5825 RETURN
5830 REM wizualizacja wynikow
5835 PRINT ' INVERSE 1;"PNP";e(x
,l,1); INVERSE 0;TAB 6;" B=";e(x
,l,2);TAB 12;" K=";e(x,l,3);TAB
18;" E=";e(x,l,4)
5836 LET m1=3: LET m2=4: GO SUB
6400: PRINT 'TAB 6;"b = ";e(x,l,
5);" [V/V]";TAB 5;"Ube= -0.65 V"
;TAB 5;"Uce= ";m;" V";TAB 6;"Ib=
";t(1,FN p(lp));" A";TAB 6;"Ic=
";e(x,l,5)*t(1,FN p(lp));" A";T
AB 6;"Ie= ";t(1,FN p(lp))*(1+e(x
,l,5));TAB 6;"P = ";-(.65+m*e(x,
l,5))*t(1,FN p(lp));" W"
5840 LET lp=lp+1: RETURN
6000 REM [illegible]
6010 CLS
6020 GO SUB 8000+100*(x-1)
6030 GO SUB 9800
6040 GO SUB 9990
6050 IF x$="P" AND i(x)>0 THEN L
ET k=0: GO SUB 9600: IF e>0 THEN
 GO TO 6070
6051 IF x$="d" AND i(x)<10 THEN
LET i(x)=i(x)+1: LET e=i(x): GO
TO 6070
6052 IF x$="s" AND i(x)>0 THEN G
O SUB 9600: IF e>0 THEN GO SUB 8
050+100*(x-1): GO TO 6000
6053 IF x$="k" THEN RETURN
6060 BEEP .1,20: GO TO 6040
6070 GO SUB 8030+100*(x-1): GO T
O 6000
6100 REM [illegible]
6105 IF lz+k>maxn THEN LET stop=
1: PRINT "blad-przekroczono rozm
iar macierzy": RETURN
6110 FOR j=2 TO k: IF e(x,l,j)>0
 THEN IF t(2,e(x,l,j))=0 THEN LE
T t(2,e(x,l,j))=lz: LET lz=lz+1
6115 NEXT j: RETURN
6200 REM [illegible]
6205 IF lp+k>maxn/2 THEN LET sto
p=1: PRINT "blad-przekroczono il
osc pradow": RETURN
6210 FOR j=1 TO k: LET t(2,lp+10
)=lz: LET lp=lp+1: LET lz=lz+1:
NEXT j: RETURN
6300 REM [illegible]
6310 CLS : GO SUB 7500: LET stop
=0: LET lz=1: LET lp=1: PRINT "a
naliza ukladu rozpoczeta"''
6350 FOR x=1 TO 9: FOR l=1 TO i(
x): GO SUB 5000+100*(x-1): NEXT
l: NEXT x: IF stop=1 THEN PRINT
''"bledy w ukladzie-koniec analiz
y": GO TO 6395
6351 IF lz=1 THEN PRINT "brak uk
ladu - koniec analizy": GO TO 63
95
6355 PRINT ''"analiza ukladu zako
nczona"''"analiza numeryczna roz
poczeta"
6360 GO SUB 1000: IF m=0 THEN GO
 TO 6395
6362 PRINT ''"napiecia w wezlach
ukladu:": FOR x=1 TO maxn/2: IF
t(2,x)=0 THEN GO TO 6364
6363 PRINT "wezel ";x;" = ";t(1,
t(2,x));" V"
6364 NEXT x
6365 LET lp=2: FOR x=1 TO 9: FOR
 l=1 TO i(x): GO SUB 5030+100*(x
-1): NEXT l: NEXT x
6370 PRINT #0;"czy powtorzyc wyn
iki? (t/n)": PAUSE 1: PAUSE 0: I
NPUT "": IF INKEY$="t" THEN GO T
O 6362
6375 RETURN
6395 PRINT #0;"nacisnij dowolny
klawisz": PAUSE 1: PAUSE 0: INPU
T "": RETURN
6400 REM [illegible]
6410 LET x1=0: IF e(x,l,m1)>0 TH
EN LET x1=t(1,FN u(m1))
6420 LET x2=0: IF e(x,l,m2)>0 TH
EN LET x2=t(1,FN u(m2))
6430 LET m=x1-x2: RETURN
7000 REM [illegible]
7010 DIM e(9,10,7): DIM i(10)
7020 RETURN
7500 REM [illegible]
7510 DIM n(maxn,maxn): DIM b(max
n): DIM t(2,maxn): DIM x(maxn)
7520 RETURN
8000 REM [illegible]
8010 REM lista
8015 PRINT "Stale Zrodla Pradowe
": PRINT
8020 PRINT "nr element I+ I- I[A
]"
8025 LET x$="NZP": GO SUB 9110:
RETURN
8030 REM edycja elementu
8040 LET k=0: GO SUB 9500
8041 LET x$="nr wezla I+": GO SU
B 9500
8042 LET x$="nr wezla I-": GO SU
B 9500
8043 LET x$="prad zrodla w [A]":
 GO SUB 9550
8045 RETURN
8050 REM kasowanie elementu
8055 LET l=4
8060 GO SUB 9400
8065 RETURN
8100 REM [illegible]
8110 REM lista
8115 PRINT "Stale Zrodla Napieci
owe": PRINT
8120 PRINT "nr element U+ U- U[V
]"
8125 LET x$="NZN": GO SUB 9110:
RETURN
8130 REM edycja elementu
8140 LET k=0: GO SUB 9500
8141 LET x$="nr wezla U+": GO SU
B 9500
8142 LET x$="nr wezla U-": GO SU
B 9500
8143 LET x$="napiecie zrodla w [
V]": GO SUB 9550
8145 RETURN
8150 REM kasowanie elementu
8155 LET l=4
8160 GO SUB 9400
8165 RETURN
8200 REM [illegible]
8210 REM lista
8215 PRINT "Zrodla Pradowe Ster.
 Napieciowo": PRINT
8220 PRINT "nr element S+ S- U+
U- Wy[S]"
8225 LET x$="ZPSN": GO SUB 9130:
 RETURN
8230 REM edycja elementu
8240 LET k=0: GO SUB 9500
8241 LET x$="nr wezla S+": GO SU
B 9500
8242 LET x$="nr wezla S-": GO SU
B 9500
8243 LET x$="nr wezla U+": GO SU
B 9500
8244 LET x$="nr wezla U-": GO SU
B 9500
8245 LET x$="wspolczynnik R[Om]"
: GO SUB 9550
8246 RETURN
8250 REM kasowanie elementu
8255 LET l=6
8260 GO SUB 9400
8265 RETURN
8300 REM [illegible]
8310 REM lista
8315 PRINT "Zrodla Pradowe Ster.
 Pradowo": PRINT
8320 PRINT "nr element S+ S- I+
I- Wk[]"
8325 LET x$="ZPSP": GO SUB 9130:
 RETURN
8330 REM edycja elementu
8340 LET k=0: GO SUB 9500
8341 LET x$="nr wezla S+": GO SU
B 9500
8342 LET x$="nr wezla S-": GO SU
B 9500
8343 LET x$="nr wezla I+": GO SU
B 9500
8344 LET x$="nr wezla I-": GO SU
B 9500
```

```
8345 LET x$="wspolczynnik k[]":
GO SUB 9550
8346 RETURN
8350 REM kasowanie elementu
8355 LET l=6
8360 GO SUB 9400
8365 RETURN
8400 REM [illegible]
8410 REM lista
8415 PRINT "Zr. Napieciowe Ster.
 Napieciowo": PRINT
8420 PRINT "nr element S+ S- U+
U- Wt[]"
8425 LET x$="ZNSN": GO SUB 9130:
 RETURN
8430 REM edycja elementu
8440 LET k=0: GO SUB 9500
8441 LET x$="nr wezla S+": GO SU
B 9500
8442 LET x$="nr wezla S-": GO SU
B 9500
8443 LET x$="nr wezla U+": GO SU
B 9500
8444 LET x$="nr wezla U-": GO SU
B 9500
8445 LET x$="wspolczynnik t[]":
GO SUB 9550
8446 RETURN
8450 REM kasowanie elementu
8455 LET l=6
8460 GO SUB 9400
8465 RETURN
8500 REM [illegible]
8510 REM lista
8515 PRINT "Zrodla Napieciowe St
er. Pradowo": PRINT

8520 PRINT "nr element S+ S- I+
I- Wr[Om]"
8525 LET x$="ZNSP": GO SUB 9130:
 RETURN
8530 REM edycja elementu
8540 LET k=0: GO SUB 9500
8541 LET x$="nr wezla S+": GO SU
B 9500
8542 LET x$="nr wezla S-": GO SU
B 9500
8543 LET x$="nr wezla I+": GO SU
B 9500
8544 LET x$="nr wezla I-": GO SU
B 9500
8545 LET x$="wspolczynnik Y[S]":
 GO SUB 9550
8546 RETURN
8550 REM kasowanie elementu
8555 LET l=6
8560 GO SUB 9400
8565 RETURN
8600 REM [illegible]
8610 REM lista
8615 PRINT "Rezystory": PRINT
8620 PRINT "nr element W+ W- R[O
m]"
8625 LET x$="R": GO SUB 9110: RE
TURN
8630 REM edycja elementu
8640 LET k=0: GO SUB 9500
8641 LET x$="nr wezla W+": GO SU
B 9500
8642 LET x$="nr wezla W-": GO SU
B 9500
8643 LET x$="rezystancja R[Om]":
 GO SUB 9550
8644 RETURN
8650 REM kasowanie elementu
8655 LET l=4
8660 GO SUB 9400
8665 RETURN
8700 REM [illegible]
8710 REM lista
8715 PRINT "Tranzystory NPN": PR
INT
8720 PRINT "nr element B  K  E
b[A/A]"
8725 LET x$="NPN": GO SUB 9120:
RETURN
8730 REM edycja elementu
8740 LET k=0: GO SUB 9500
8741 LET x$="nr wezla B": GO SUB
 9500
8742 LET x$="nr wezla K": GO SUB
 9500
8743 LET x$="nr wezla E": GO SUB
 9500
8744 LET x$="wzmocnienie b[A/A]"
: GO SUB 9550
8745 RETURN
8750 REM kasowanie elementu
8755 LET l=5
8760 GO SUB 9400
8765 RETURN
8800 REM [illegible]
8810 REM lista

8815 PRINT "Tranzystory PNP": PR
INT
8820 PRINT "nr element B  K  E
b[A/A]"
8825 LET x$="PNP": GO SUB 9120:
RETURN
8830 REM edycja elementu
8840 LET k=0: GO SUB 9500
8841 LET x$="nr wezla B": GO SUB
 9500
8842 LET x$="nr wezla K": GO SUB
 9500
8843 LET x$="nr wezla E": GO SUB
 9500
8844 LET x$="wzmocnienie b[A/A]"
: GO SUB 9550
8845 RETURN
8850 REM kasowanie elementu
8855 LET l=5
8860 GO SUB 9400
8865 RETURN
9000 REM [illegible]

9010 GO SUB 9900
9020 GO SUB 9990
9030 IF x$>="0" AND x$<="8" THEN
 LET x=VAL x$+1: CLS : GO SUB 60
00: GO TO 9000
9040 IF x$="w" THEN GO SUB 6300:
 GO TO 9000
9041 IF x$="n" THEN PAUSE 1: PRI
NT #0;"nowy uklad? (t/n)": PAUSE
 0: INPUT "": IF INKEY$="t" THEN
 GO SUB 7000: GO TO 9000
9042 IF x$="l" THEN CLS : PRINT
"lista elementow ukladu": FOR x=
1 TO 9: GO SUB 8025+100*(x-1): N
EXT x: PRINT : PRINT #0;"Nacisni
j dowolny klawisz": PAUSE 1: PAU
SE 0: INPUT "": GO TO 9000
9045 BEEP .1,20
9050 GO TO 9020
9100 REM [illegible]
9110 REM 2 wezly,1 parametr
9115 FOR k=1 TO i(x): PRINT k;TA
B 3;x$;e(x,k,1);TAB 11;e(x,k,2);
TAB 14;e(x,k,3);TAB 17;e(x,k,4):
 NEXT k: RETURN
9120 REM 3 wezly,1 parametr
9125 FOR k=1 TO i(x): PRINT k;TA
B 3;x$;e(x,k,1);TAB 11;e(x,k,2);
TAB 14;e(x,k,3);TAB 17;e(x,k,4);
TAB 20;e(x,k,5): NEXT k: RETURN
9130 REM 4 wezly,1 parametr
9135 FOR k=1 TO i(x): PRINT k;TA
B 3;x$;e(x,k,1);TAB 11;e(x,k,2);
TAB 14;e(x,k,3);TAB 17;e(x,k,4);
TAB 20;e(x,k,5);TAB 23;e(x,k,6):
 NEXT k: RETURN
9400 REM [illegible]
9410 LET i(x)=i(x)-1: IF e=10 TH
EN RETURN
9420 FOR k=e TO i(x): FOR j=1 TO
 l
9430 LET e(x,k,j)=e(x,k+1,j)
9440 NEXT j: NEXT k: RETURN
9500 REM [illegible]

9505 IF k=0 THEN LET x$="numer u
kladowy elementu"
9510 LET k=k+1
9515 INPUT (x$);": ";e(x,e,k)
9520 IF FN z(k=1,maxw,e(x,e,k))
THEN GO TO 9515
9530 RETURN
9550 REM [illegible]
9560 LET k=k+1: INPUT (x$);": ";
e(x,e,k)
9565 RETURN
9600 REM [illegible]
9610 INPUT "padaj nr elementu: "
;e: IF FN z(0,i(x),e) THEN GO TO
 9610
9620 RETURN
9800 REM [illegible]
9810 PRINT AT 15,11; INVERSE 1;"
FUNKCJE:"
9820 PRINT
9830 IF i(x)>0 THEN PRINT TAB 5;
"p-poprawienie elementu"


9831 IF i(x)<10 THEN PRINT TAB 5
;"d-dolaczenie elementu"
9832 IF i(x)>0 THEN PRINT TAB 5;
"s-skasowanie elementu"
9833 PRINT TAB 5;"k-koniec edycj
i zbioru"
9840 RETURN
9900 REM [illegible]
9910 CLS : PRINT TAB 7; INVERSE
1;"FUNKCJE EDYTORA:"
9915 PRINT
9920 PRINT "0-Stale Zrodlo Prado
we"
9921 PRINT "1-Stale Zrodlo Napie
ciowe"
9922 PRINT "2-Zr. Pradowe Ster.
Napieciowo"
9923 PRINT "3-Zr. Pradowe Ster.
Pradowo"
9924 PRINT "4-Zr. Napiecowe Ster
. Napieciowo"
9925 PRINT "5-Zr. Napieciowe Ste
r. Pradowo"
9926 PRINT "6-Rezystor"
9927 PRINT "7-Tranzystor NPN"
9928 PRINT "8-Tranzystor PNP"
9930 PRINT : PRINT "w-wykonanie
obliczen"
9931 PRINT "n-nowy uklad"
9932 PRINT "l-lista elementow uk
ladu"
9940 RETURN
9990 REM [illegible]
9991 PRINT #0;AT 0,8;"wybierz fu
nkcje"
9992 PAUSE 0: LET x$=INKEY$: IF
x$<>"" THEN INPUT "": RETURN
9993 GO TO 9991
9998 REM [illegible]
9999 CLEAR : SAVE "analiza DC" L
INE 1
```

Zgadywanka — gra

```
#include <stdio.h>

void main()
{
int wzorzec,liczba,licznik;
printf ("liczbe wzorcowa podaje pierwsza ");
printf (" osoba, druga ja zgaduje\n\n");
do
{
   printf("podaj wzorzec <0-999>: ");
   wzorzec=dana ();
   system ("cls");
   for (licznik=1;licznik<=20;licznik++)
   {
      printf("trafiasz %d raz (wpisz liczbe <0-999>): ",
      licznik);
      liczba=dana ();
      if (liczba==wzorzec)
         {
         printf(" - trafiles!!!\n");
         goto end;
         }
         else
         {
         if (liczba>wzorzec)
          printf(" - liczba za duza\n");
          else
          printf(" - liczba za mala\n");
         }
      }
   printf("\ntrafiasz i trafiasz, az do znudzenia!.");
   printf(" Przerywam program!");
   break;
   end:
   printf("\nczy grasz jeszcze raz? (t/n)\n\n");
   }
while (getch()=='t');
}

dana () /*wczytanie liczby 3-cyfrowej*/
{
   int licznik,liczba=0;
   char znak;
   #define maxchar 3
   #define enter 13
   for (licznik=1;licznik<=maxchar;licznik++)
   {
      do
      {
         znak=getch();
         if ((znak==enter)&&(licznik>1))
            goto end;
      }
      while((znak<'0')||(znak>'9'));
      putchar(znak);
      liczba=liczba*10+znak-'0';
   }
end:
return(liczba);
}
```

Jak oszczędnie przechowywać ekrany na dyskietce

MIECZYSŁAW SKONIECZNY, PIOTR RAKOWSKI

Obraz wyświetlany na ekranie mikrokomputera AMSTRAD CPC 6128 zajmuje 17 KB pamięci począwszy od adresu C000. Można go przechowywać na dyskietce, gdzie zajmuje około 10% zawartości jednej strony. Chcemy przedstawić programy do kompresji i dekompresji obrazów. Używając ich musimy ograniczyć wielkość pamięci przeznaczonej na BASIC do 16383 bajtów (instrukcja MEMORY 16383). Pierwszy z omawianych programów służy do kompresji ekranu. Skompresowany obraz jest umieszczony w pamięci od adresu 4002, a jego długość (potrzebna do nagrania na dyskietkę) można odczytać pod adresem 4000. Kompresja obrazu jest realizowana przez zastąpienie ciągu identycznych bajtów dwoma przedstawiającymi ich wartość i długość ciągu. Program ten jest relokowalny w pamięci mikrokomputera. Wywołuje się go instrukcją:

```
10 'kod maszynowy  "kompres"
20 adres=38500:ilosclinii=5
30 RESTORE 200
40 FOR i=1 TO ilosclinii
50 s=0:READ a$,w$
60 FOR j=1 TO 29 STEP 2
70 a=VAL("&"+MID$(a$,j,2))
80 POKE adres,a
90 s=s+a:adres=adres+1
100 NEXT j
110 IF s<>VAL(w$) THEN PRINT"Blad w linii nr ";190+i*10:STOP
120 NEXT i
130 SAVE"kompres",b,38500,65
200 DATA 1102402100c00ec80650c5e50e007e,1174
210 DATA 0c2305280abe28f8127913121318ee,1037
220 DATA 1279131213e17cfef828023809d638,1423
230 DATA 67015000091804010008090c10d20ce,683
240 DATA eb220040c90000000000000000000,534
```

CALL adres

gdzie „adres" jest wartością, od której nagramy program do pamięci. Wersja źródłowa programu jest następująca:

Drugi z programów służy do dekompresji obrazów utworzonych poprzednio programem „kompres". Aby otrzymać obraz na ekranie należy wgrać do pamięci mikrokomputera od adresu 4000 skompresowany obrazek i wywołać program „dekompres" instrukcją:

CALL adres

gdzie „adres„ jest wartością, od której nagrywamy program do pamięci. Wersja źródłowa tego programu jest następująca:

```
10 'kod maszynowy  "dekompres"
20 adres=38600:ilosclinii=4
30 RESTORE 200
40 FOR i=1 TO ilosclinii
50 s=0:READ a$,w$
60 FOR j=1 TO 29 STEP 2
70 a=VAL("&"+MID$(a$,j,2))
80 POKE adres,a
90 s=s+a:adres=adres+1
100 NEXT j
110 IF s<>VAL(w$) THEN PRINT"Blad w linii nr ";190+i*10:STOP
120 NEXT i
130 SAVE"dekompre",b,38600,58
200 DATA 2102401100c00ec8d50650c57e234e,1257
210 DATA 231213050d20fa78b720f2c1d1c57a,1670
220 DATA fef82802380bd63857015000eb09eb,1528
230 DATA 1806010008eb09ebc10d20cfc90000,1164
```

Program demonstracyjny napisany w języku BASIC przedstawia, jak należy wykonać kompresję (linie 120—360) i dekompresję obrazu (linie 380—520). Kompresowane obrazy są nagrywane na dyskietkę z rozszerzeniem CMP. Dekompresja jest realizowana także w zbiorach z rozszerzeniem CMP. Wersja źródłowa tego programu jest następująca:

```
10 'program demonstracyjny
20 'kompresja i dekompresja ekranu
30 '
40 MEMORY 16383
50 MODE 1:WINDOW £3,12,28,13,18:PAPER £3,1:PEN £3,0:CLS £3
60 PRINT£3:PRINT£3," 1 - kompresja"
70 PRINT£3," 2 - dekompresja"
80 PRINT£3:PRINT£3,"     Wybierz"
90 a$=INKEY$:IF a$<>"1" AND a$<>"2" THEN 90
100 IF a$="2" THEN 380
110 '
120 'kompresja
130 MODE 2
140 LOAD"kompres
150 WINDOW £2,20,60,21,25:PAPER £2,1:PEN £2,0
160 CLS£2
170 PRINT£2:PRINT£2," Wloz dyskietke z obrazkiem do kompresji"
180 PRINT£2," i nacisnij dowolny klawisz."
190 CALL &BB06
200 CAT
210 CLS£2
220 PRINT£2:PRINT£2," Nazwa obrazka do kompresji :";:INPUT£2,"", a$
230 PRINT£2," Nazwa obrazka po kompresji :";:INPUT£2,"",b$
240  LOAD a$+".scr",&C000
250 CALL 38500
260 CLS:CLS£2
270 PRINT£2
280 PRINT£2,"   Zmien dyskietke i nacisnij dowolny"
290 PRINT£2,"   klawisz ,aby nagrac skompresowany "
300 PRINT£2,"   obrazek."
310 CALL &BB06
320 DL=PEEK(&4000)+256*PEEK(&4001)-16385
330 SAVE b$+".cmp",B,&4002,DL
340 CLS£2:PRINT£2:PRINT£2,"   Nastepny obrazek ? :";:INPUT£2,"",a$
350 IF a$="t" OR a$="T" THEN 160
360 RUN
370 '
380 'dekompresja
390 MODE 2:LOAD"dekompre
400 WINDOW £1,20,60,21,25:PAPER £1,1:PEN £1,0
410 CLS£1:PRINT£1
420 PRINT£1,"     Wloz dyskietke z obrazkiem do "
430 PRINT£1,"     dekompresji i nacisnij dowolny"
440 PRINT£1,"     klawisz."
450 CALL &BB06
460 CLS£1:CAT:PRINT£1:PRINT£1," Nazwa obrazka do dekompresji :";:INPUT£1,"",a$
470 PRINT£1:PRINT£1," W jakim trybie ?:";:INPUT£1,"",tryb
480 MODE tryb
490 LOAD a$+".cmp",&4002
500 CALL 38600
510 FOR i=1 TO 10000:NEXT
520 RUN
```

SYSTEMY OPERACYJNE I INNE GOTOWE PROGRAMY

O sprzęcie powiedzieliśmy sobie już bardzo dużo, w każdym razie wiele jak na **Podstawy Informatyki**. Niestety w tym przypadku ilość nie "przechodzi" w jakość. Urządzenia te, choćby wszystkie zgromadzone razem w jednym miejscu będą po prostu górą (zresztą pokaźną) złomu metali kolorowych, sztucznego tworzywa, szkła i innych składowych tego co potocznie nazywamy **komputerem**.

Po "ożywieniu" sprzęt ten, jak niegdyś, *Frankenstein*, natychmiast przystępuje do działania. Szczęśliwie komputery nie zachowują się jak dziecko wyobraźni pani Shelly, są bardzo "łagodne", cierpliwe - po prostu są *życzliwe*. Jest to możliwe dzięki specjalnej formie owego *ożywienia*. Polega ono na podaniu energii (w Europie najlepiej 220V i 50 Hz) oraz wprowadzeniu do pamięci maszyny odpowiedniego programu. Pamiętając o naszych doświadczeniach z programowaniem (IKS nr 11/88) nie trudno jest sobie wyobrazić ogrom pracy, jaki trzeba włożyć w pisanie programów, które zapewnią prawidłowe działanie tysięcy komputerów.

Wróćmy na chwilę do *monstrów*, ktore współcześnie nam "obrodziły", na przykład w filmach "Superglina" i "Elektroniczny morderca". Ożywienie tych postaci polegało na przekazaniu naszym bohaterom zdolności (czytaj programów) poruszania kończynami, widzenia i analizy obrazu, słyszenia i rozumienia mowy ludzkiej . Poza sposobem realizacji poleceń otrzymywanych przez owe istoty, gdy uwidacznia się ich indywidualność (na przykład czarny charakter) pozostałe zdolności są typowe, niezmienne. Czy był to Elektroniczny morderca, czy Superglina - istoty te wykonywały te same czynności. Jeśli okazało by się, że są to komputery, a było tak w istocie (prawie), wystarczyło powielać ten sam zakodowany w ich *sztucznym mózgu (komputerze)* program, by standardowe zachowania owych postaci były realizowane. Nie trzeba zatem płacić ogromnych sum za *oprogramowanie* kolejnych egzemplarzy naszych bohaterów - przynajmniej w części dotyczącej ich podstawowych zachowań, jakimi w tym przypadku, poza chodzeniem, bieganiem, była nadzwyczajna celność strzelania z broni palnej. Można bowiem powielać program sterujący wykonywaniem typowych czynności... komputera w każdej podobnej maszynie.

Te *realistyczne* przykłady mają na celu zwrócenie uwagi na ideę konstrukcji tego typu programów:

a) wykonywanie często powtarzanych przez użytkownika czynności,

b) możliwość wykorzystania określonej wersji programu (systemu) w wielu komputerach.

Kupując zatem nowy komputer, nie musimy budować od nowa programu, który na przykład *przejrzy* nam dysk twardy i wyświetli jego zawartość, w czytelnej dla nas postaci. Podobnie przydałby się programik, który automatycznie sprawdzi dyskietkę, czy jest dobra (bez *dziur*) i co na niej zapisano. Mało tego, nawet ruch głowicy po powierzchni dysku też trzeba *oprogramować*. Oznacza to, że każdą najdrobniejszą czynność wykonywaną przez maszynę należało wcześniej przewidzieć i opisać (w języku wewnętrznym komputera). Zważywszy zatem na wspomnianą uniwersalność omawianych operacji napisano podprogramy (moduły), które złożone w całość stanowią o podstawowych *zdolnościach* komputera. Nie jest to oczywiście *chodzenie*, a na przykład czytanie (z dysku). Wspomniana całość to **SYSTEM OPERACYJNY** - program, który wykonuje wszystkie standardowe (typowe) funkcje powierzane maszynie. Oczywiście mamy bardzo wiele typów systemów operacyjnych - małych i dużych.

Przyjrzyjmy się najpopularniejszemu (na pewno u nas) - DOSowi - Dyskowy Operacyny System dla mikrokomputerów PC. Jego nazwa wskazuje, że program ten powstał głównie po to, by usprawnić wykorzystywanie dysków (twardych i elastycznych), i tak było faktycznie. Ale *przy okazji* nie pominięto zainstalowania tam modułów wspomagających inne bardzo ważne zdolności mikrokomputera.

Zważywszy na to, że system operacyjny jest tylko programem, po każdym włączeniu komputera DOS musi być od nowa wczytywany z dysku (do RAM). *Ktoś* to musi zrobić, w związku z tym, że RAM jest pamięcią ulotną (zobacz IKS nr 12/88), w maszynie, po włączeniu sieci, nie ma żadnego programu. Stanowiło to o konieczności zainstalowania w komputerze takiej pamięci (ROM lub RAM zasilanej bateryjką), która nie *zgubi* (po wyłączeniu sieci) programu, *potrafiącego* wczytać DOSa. Jest to **BIOS** - program automatycznie uruchamiany zawsze po włączeniu maszyny do sieci.

BIOS *zajmuje się* także obsługą operacji związanych z wprowadzaniem i wyprowadzaniem innych (poza DOSem) informacji, a także automatycznym testowaniem podstawowych układów komputera. Innymi słowy program ten (wspomagany przez programy MSDOS.SYS oraz IO.SYS) np.: automatycznie wpisuje *trudne* liczby (np.: 1010101) do pamięci RAM następnie odczytuje je by ostatecznie porównać to co zostało zapisane z rezultatem odczytania. Proces

owej kontroli jest sygnalizowany na ekranie monitora - ukazują się tam kolejne adresy (liczby) testowanych komórek pamięci. Wynik porównania różny od zera oznacza, że określona kostka pamięci pracuje błędnie i należy ją wymienić. Tak wygląda w skrócie testowanie pamięci, od którego rozpoczyna pracę każdy komputer PC. Testowanie stanowi przykład standardowych czynności maszyny. Stanowi to, dla nas użytkowników, ciągle odnawiany kredyt zaufania co do jakości usług oferowanych przez komputer. Czyli raz napisany test powtarzany jest codziennie w tysiącach mikrokomputerów określonego typu, np.: w IBM PC/XT.

Podobnie jest z procesem (a jest on dosyć złożony) *obsługi* klawiatury czy monitora. Wykorzystywany jest do tego celu specjalny program, który *rozpozna* naciskany przez nas klawisz, umieści jego kod w pamięci, pobierze odpowiedni wzorzec (mozaikę punktów) tworzący znak i ostatecznie wyśle tę mozaikę do monitora. Na szczęście wszystkie te czynności wykonywane są przez oprogramowanie systemowe, jesteśmy zwolnieni od troski o te podstawowe zdolności maszyny.

Oczywiście jest to drobna część uniwersalnych czynności komputera. DOS realizuje ich znacznie więcej. Na przykład dostałem (kupiłem, załatwiłem) dyskietkę z doskonałym opisem edytora testów pod nazwą IKSEDYT.CHI. Niestety dyskietkę muszę za kwadrans oddać. Nie ma problemu - wkładam dyskietkę do napędu oznaczonego literą A i na wszelki wypadek polecam DOSowi by sprawdził, czy aby na pewno jest tam zapisany zbiór (nasz opis edytora). Wystarczy napisać:

DIR

i na ekranie ukazują się nam wszystkie zapisane na dyskietce zbiory,np.:

```
IKSEDYT    CHI    30000 19/01/89    9:55
BAJBAJ     CHI    30000 18/01/89   10:11
KOMKOM     CHI   120000 10/12/88    9:93
      File(s)    180000 bytes free
```

Okazuje się, że mam ostatnią dyskietkę, na której może być trochę miejsca. Wkładam ją do napędu B i przekazuję DOSowi, aby teraz zaczął interesować się napędem B pisząc B: i powtarzam znane już nam polecenie DIR W odpowiedzi na ekranie ukazuje się:

```
TEST1    TXT   100000  19/12/88   11:12
TEST2    TXT   200000  10/10/87   12:11
TEST3    TXT    50000   8/08/88    4:38
NOWA     TXT   150000   7/11/88   10:23
      4 File(s)     310000 bytes free
```

Powyższy komunikat oznacza, że są tam zapisne cztery zbiory (files) i wolne jest aż 310 000 bajtów (bytes free). Wróćmy do naszego remanentu na dyskietce w napędzie A. Okazuje się, że interesujący nas zbiór IKSEDYT.CHI wymaga tylko 30 000 bajtów pamięci (druga kolumna). Zatem jest dostatecznie dużo pamięci na mojej dyskietce. Wobec tego kopiuję interesujący mnie zbiór z napędu A na B.

```
B:\> COPY A:IKSEDYT.CHI
```

w odpowiedzi maszyna pisze:

```
1 FILE(S) COPIED
```

Oznacza to, że kopiowanie zostało zakończone. Na wszelki wypadek sprawdzam, czy operacja została poprawnie wykonana.

DIR

Na ekranie monitora ukazuje się:

```
TEST1    TXT   100000  19/12/88   11:12
TEST2    TXT   200000  10/10/87   11:11
TEST3    TXT    50000   8/08/88    4:08
NOWA     TXT   150000   7/11/88   10:00
IKSEDYT  CHI    30000  19/01/89    9:55
      5 File(s)     280000 bytes free
```

W związku z tym, że była to jednak tylko demonstracja - szybko kasuję efekt dotychczasowej pracy i sprawdzam skuteczność *mazania*;

```
B:\> ERASE IKSEDYT.CHI
B:\> DIR
TEST1    TXT   100000  19/12/88   11:12
TEST2    TXT   200000  10/10/87   12:11
TEST3    TXT    50000   8/08/88    4:38
NOWA     TXT   150000   7/11/88   10:23
      4 File(s)     310000 bytes free
```

Kończąc prezentowane przykłady, przy okazji wyświetlania zawartości dyskietek, pragnę zwrócić uwagę na bardzo użyteczne informacje, które zawarte są w dwóch ostatnich kolumnach *remanentów* dyskietek, są to: data i czas założenia (zbudowania) danego zbioru.

Poza przedstawionymi przykładami DOS *potrafi* także między innymi: zmieniać pamiętaną przez maszynę datę i czas zegarowy, kasować, drukować, wyświetlać i szukać wybrane informacje, układać *po kolei* czyli *sortować* np.: alfabetycznie. Pełna lista *zdolności* DOSa z krótkim opisem, jak je wykorzystać, to dokumentacja w postaci pokaźnego tomu.

Jak już kilkakrotnie wspominałem System Operecyjny - w naszym przykładzie DOS - *świadczy* podstawowe, niezbędne każdemu użytkownikowi *usługi*. Wydaje się, że można go przyrównać do solidnego fundamentu. Konstrukcja wznoszonego na nim budynku może być różna - zależna od praktycznego przeznaczenia owego dzieła. Na przykład dla samochodziarzy będzie to garaż, dla handlowców - hala targowa, dla Dyrekcji PKS - dworzec autobusowy.

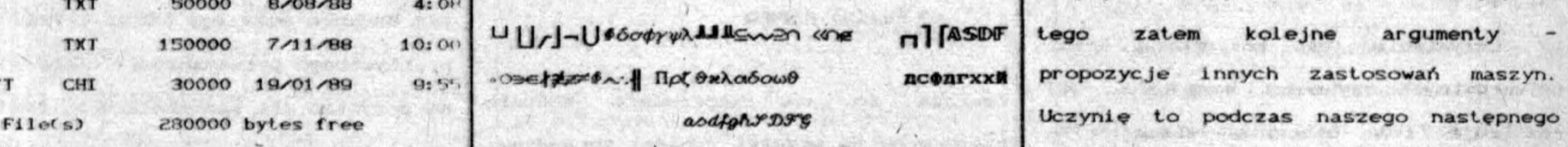
Wróćmy jednak do komputera. Nasz fundament to system operacyjny, budowla - narzędziowe i użytkowe programy. Istnieją ich tysiące. Oznacza to, że kupujemy komputer wraz z systemem operacyjnym, a dokupienie programu dostosowanego do naszych specyficznych potrzeb nie stanowi problemu. W wielu przypadkach (chyba w większości) użytkownicy sprzętu PC nie muszą tworzyć nowego, dostosowanego do ich potrzeb programu. Oznacza to, że niemal na drugi dzień po kupieniu komputera i odpowiedniego oprogramowania możemy wykorzystywać maszynę do wykonywania (wspomagania) określonych prac.

Bardzo użyteczne okazały się tak zwane programy narzędziowe. Jest ich bardzo wiele (NORTON, PC TOOLS, XTREE i inne). Każdy z nich to kolejny krok w uproszczeniu obsługi komputera. Wykonują one bowiem wszystkie funkcje DOSa i jeszcze sporo innych. Wywołanie, uruchomienie owych funkcji zazwyczaj polega na przesuwaniu kursora po ekranie lub napisaniu jednej litery wyróżniającej określone polecenie. Na monitorze wyświetlane jest bowiem kilkadziesiąt, ładnie ułożonych w tabelkach, słów i zwrotów (niestety zazwyczaj w języku angielskim). Na przykład: DYSK, DRUKUJ, KASUJ, KOPIUJ. Wystarczy kursorem *dotknąć* słowo DYSK lub napisać literkę D (gdy wybieram drukowanie piszę znak R, kasowanie - znak K i odpowiednio - O dla kopiowania) ,a na ekranie wyświetlą się nam literki - *imiona* napędów dyskowych: A, B, C. *Dotykamy* A i na ekranie wyświetlona zostaje lista zbiorów, które są zapisane na dyskietce (aktualnie włożonej do napędu A). Podobnie jak miało to miejsce przy wykorzystaniu instrukcji DIR w DOSie.

```
TEST1       TXT      100000  19/12/88    11:1
TEST2       TXT      200000  10/10/87    11:11
TEST3       TXT       50000   8/08/88     4:08
NOWA        TXT      150000   7/11/88    10:00
IKSEDYT     CHI       30000  19/01/89     9:5
      5 File(s)      280000 bytes free
```

Po co zatem ten program? Znakomicie usprawnia (przyspiesza) wykonywanie operacji *narzędziowych*, choćby tylko przez to, że wystarczy *dotknąć* dane polecenie (a nie napisać) i natychmiast jest ono realizowane. Wróćmy do naszego przykładu. Po wyświetleniu zawartości dyskietki, także tym razem *dotykam* wybrany zbiór i oznaczam go przyciskając klawisz F1 (klawisz funkcyjny). Na przykład zbiór **NOWA** zostaje podświetlony (*napisany tłustym drukiem*). Teraz wybieram jedną z funkcji: KOPIUJ, KASUJ, DRUKUJ - w zależności od tego co zamierzam wykonać. Oczywiście lista dostępnych operacji jest znacznie większa, a opisany, przykładowy, program tylko w drobnej części ilustruje możliwości omawianych narzędzi. Ich umiejętne wykorzystanie znacznie ułatwia proces współpracy z maszyną - staje się ona (praca i maszyna) coraz sympatyczniejsza.

A cóż te inne użytkowe programy mogą wykonywać? Na pewno wszystko to co było realizowane na starszym sprzęcie komputerowym! Świadom jestem tego, że systemy finansowe, kadrowe, materiałowe (a jest ich wiele) są odległe od naszych zainteresowań. Stąd też pragnę zwrócić uwagę na kilka atrakcyjniejszych dla nas zastosowań. Przyjrzyjmy się na przykład edytorowi tekstów. Dzięki temu programowi, komputer zamienia się w uniwersalną maszynę do pisania. To co napiszemy na klawiaturze zostanie wydrukowane. Nie prościej i taniej (!) kupić sobie po prostu maszynę do pisania? Przede wszystkim edytor (typowy) dysponuje bogatym zestawem krojów czcionek (np.: 20 razy 2 - bo duże i małe litery) z nieograniczoną możliwością tworzenia nowych własnych znaków - może to być cyrylica, znaki matematyczne, symbole elktroniczne lub chińskie obrazki, np.:



Wybór dowolnego kroju liter (znaków) polega na naciśnięciu jednego lub dwóch klawiszy. Mimo tych zalet cena komputera - kilka milionów - budzi nadal duże wątpliwości co do celowości takiej inwestycji.

Gdy piszemy, maszyna dba o to by każdy wiersz miał zaplanowaną przez nas liczbę znaków. Polega to na tym, że piszemy *ciurkiem* - bez zastanawiania się gdzie skończyć wiersz - kontroluje to maszyna przenosząc słowo, które powoduje przepełnienie zaplanowanej przez nas liczby znaków w wierszu do następnej linii tekstu. Aby wiersz z którego zabrano wyraz był odpowiednio długi, komputer dodaje w nim, między poszczególne słowa, odpowiednią liczbę odstępów. Marginesy są idealnie równe. Przykładem tego jest niniejszy tekst, który był pisany właśnie przy pomocy omawianego edytora. Po ukończeniu całej strony, zauważam (oglądając ją na ekranie monitora), że *zjadłem* słowo i parę liter. Przesuwając kursor w odpowiednie miejsce *wciskam* tam brakujące słowa i znaki. Itp., itp.

Jeszcze o jednej własności niektórych tekstowych edytorów należy powiedzieć parę słów - wykluczają one błędy ortograficzne! Porównują każde pisane przez nas słowo z pamiętanym słownikiem ortograficznym - w efekcie komputer czasami buczy, *oj buczy!*

Opisałem tutaj zaledwie drobną cześć dostępnych możliwości edytora tekstów. Mimo ich niewątpliwej atrakcyjności (także użyteczności) - proponowane zastosowanie nie powinno stanowić koronnego argumentu podjęcia decyzji o zakupie komputera. Dodajmy do tego zatem kolejne argumenty - propozycje innych zastosowań maszyn. Uczynię to podczas naszego następnego spotkania (to już chyba wiosną?). Przedmiotem VII części **PODSTAW INFORMATYKI** będzie nieco inny edytor - tym razem obrazów (szkoda że nasz IKS nie jest kolorowy), ciekawe programy obliczeń statystycznych oraz podzielę się wrażeniami z przebiegu łączności komputerowej, nie tylko w skali kraju.

Włodzimierz Gogołek

Program „MIERNIK"

W pracowniach laboratoryjnych naszych szkół jest szeroko wykorzystywany do pomiarów miernik uniwersalny typu UM3 (lub podobny). Szybkie i prawidłowe odczytywanie wskazań mierników tego typu wymaga wprawy poprzedzonej długotrwałym treningiem. Zastąpienie w czasie treningu przyrządu pomiarowego programem mikrokomputerowym pozwala uniknąć ryzyka jego uszkodzenia i uatrakcyjnia proces nauki.

Przedstawiany program „MIERNIK" służy spełnieniu przedstawionych wyżej celów. Został opracowany w języku Basic na mikrokomputer typu ZX Spectrum (lub podobny). Po uruchomieniu na ekranie rysowana jest podziałka miernika UM3 z literowym oznaczeniem amperomierza i woltomierza. Obrazy te są zapamiętywane do późniejszego wykorzystania. Jest to część wstępna, przygotowawcza programu. W części głównej losowany jest rodzaj wykorzystania miernika (woltomierz, amperomierz), zakres pomiarowy oraz wartość wielkości mierzonej (pamiętana przez komputer). W efekcie na ekranie ukazuje się skala przyrządu ze wskazówką w położeniu odpowiadającym wylosowanej wcześniej wartości wielkości mierzonej.

Rodzaj wielkości mierzonej (rodzaj pracy miernika) jest obrazowany przez wyświetlenie symbolu „A" lub „U" poniżej podziałki. W oknie systemowym pojawia się komunikat o wylosowanym zakresie pomiarowym i polecenie wprowadzenia odczytanej przez trenującego wartości. Obraz ten jest przedstawiony na poniższym rysunku. Wprowadzona wartość jest porównywana z zapamiętaną. Program dopuszcza popełnienie błędu odczytu nie większego niż 0,5 działki (1.5%). Wystąpienie większego błędu powoduje wyświetlenie komunikatu o nim i polecenie dokonania powtórnego odczytu. Po trzecim błędnym odczycie na ekranie ukazuje się informacja na temat prawidłowego postępowania podczas odczytywania wskazań mierników wielozakresowych. Po jej przeczytaniu ćwiczący kieruje program (przez naciśnięcie dowolnego klawisza) do początku, celem dalszego ćwiczenia.

Podanie przez trenującego wartości z dopuszczalnym błędem powoduje automatyczne powtórzenie ćwiczenia dla nowych wartości.

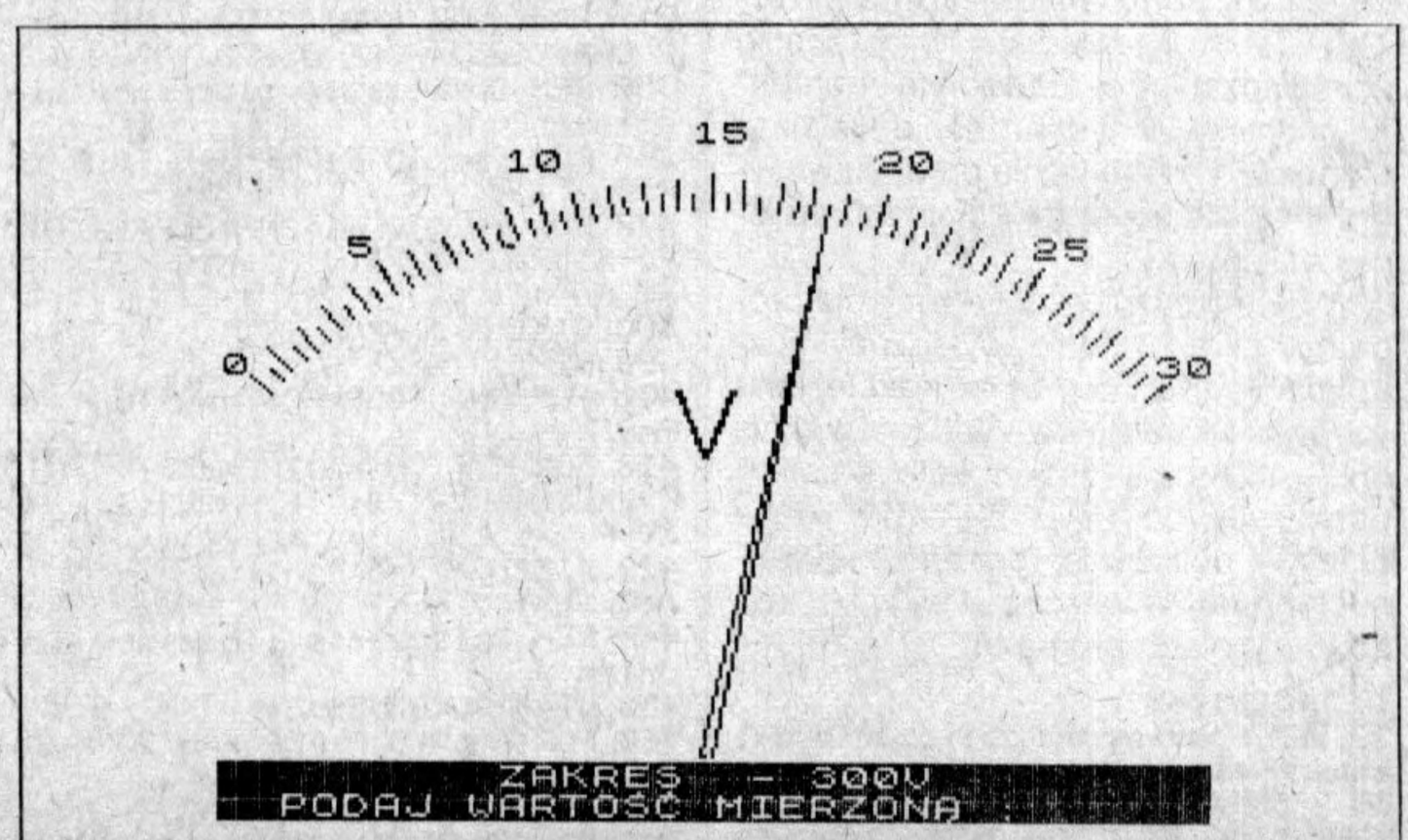
Program działa szybko dzięki zapamiętaniu obrazów z narysowanej podziałki i przenoszeniu ich za pomocą wstawki w kodzie maszynowym.

Podstawowe uwagi dotyczące korzystania z załączonego listingu programu.

1. Linie 20,30 — program w języku maszynowym do przenoszenia obrazów. Adres oryginału w komórkach 50001,50002; adres kopii w komórkach 50004,50005.
2. Linie 40,50 i 2000—2000 — rysowanie i zapamiętanie obrazów z podziałkami amperomierza i woltomierza.
3. Linie 60,70 — wczytanie do obszaru UDG polskich liter.
4. Linie 1000—1060 — podprogram wyświetlający instrukcję o prawidłowym postępowaniu podczas odczytywania wskazań miernika.

Ryszard ZIEMKOWSKI

```
 10 BORDER 0: INK 0: PAPER 7: CLEAR 49999
 20 FOR i=50000 TO 50011: READ b: POKE i,b: NEXT i
 30 DATA 33,0,64,17,56,199,1,0,27,237,176,201
 40 PLOT 119,81: DRAW 0,13: DRAW 4,3: DRAW 3,0: DRAW 4,-5: DRAW 0,-13: DRAW -1,
0: DRAW 0,13: DRAW -3,4: DRAW -3,0: DRAW -3,-4: DRAW 0,-13: DRAW 0,7: DRAW 8,0:
DRAW 0,1: DRAW -8,0: GO SUB 2000: RANDOMIZE USR 50000
 50 PRINT AT 9,14;"   ";AT 10,14;"   ";AT 11,14;"   ": PLOT 124,82: DRAW 1,0: D
RAW 7,18: DRAW -1,0: DRAW -6,-17: DRAW -1,0: DRAW -6,17: DRAW -1,0: DRAW 7,-18:
POKE 50004,144: POKE 50005,226: RANDOMIZE USR 50000: POKE 50004,0: POKE 50005,64
 60 FOR i=0 TO 167: LET a=USR "a"+i: READ b: POKE a,b: NEXT i
 70 DATA 0,0,56,4,60,68,60,6,0,60,66,66,126,66,70,8,4,8,28,32,32,32,28,0,8,60,8
2,64,64,66,60,0,0,0,56,68,120,64,60,8,0,126,64,124,64,64,126,12,16,0,124,8,16,32
,124,0,24,126,4,8,16,32,126,0,8,16,124,8,16,32,124,0,8,126,20,8,16,32,126,0,0,64
,72,80,96,64,126,0,0,16,20,24,48,16,12,0,8,74,98,82,74,70,66,0,8,16,120,68,68,68
,68,0,8,16,56,68,68,68,56,0,8,60,82,66,66,66,60,0,0,60,66,66,66,36,102,0,36,74,7
4,74,60,8,8,48,8,16,56,64,56,4,120,0,8,60,64,60,2,66,60,0,0,0,34,34,34,34,93,128
 80
100 DIM t(16): FOR i=1 TO 16: READ t(i): NEXT i
110 DATA .0015,.006,.015,.06,.15,.6,1.5,6,1.5,6,15,30,60,150,300,600
120 LET KON=49.5: LET ZAKRES=KON-POCZ
130 LET I=0: LET WSKAZ=RND*ZAKRES: LET MIER=INT (RND*16): CLS
140 IF MIER<8 THEN  POKE 50001,56: POKE 50002,199: RANDOMIZE USR 50000: GO TO 1
60
150 POKE 50001,144: POKE 50002,226: RANDOMIZE USR 50000
160 LET KAT=(POCZ+WSKAZ)/180*PI: LET DL=149: LET X0=125: LET Y0=0: PLOT X0,Y0:
DRAW -1,0: LET XC=DL*(SIN (KAT)): LET YC=DL*(COS (KAT)): DRAW XC+1,YC: DRAW -XC+
2,-YC
170 LET t$="V": IF MIER<8 THEN  LET t$="A"
180 LET t$=STR$ t(MIER+1)+t$: IF MIER<4 THEN  LET t$="0"+t$
190 INPUT AT 0,7;"ZAKRES  -  ";(t$)'"PODAJ WARTOSC MIERZONA : ";A: LET ODCZ=A*Z
AKRES/t(MIER+1)
200 LET BLAD=ABS (ODCZ-WSKAZ)
210 IF BLAD<=1.5 THEN  GO TO 130
220 LET I=I+1
230 IF I>2 THEN  GO SUB 1000: GO TO 130
240 PRINT #1;"ZŁY ODCZYT - SPROBUJ JESZCZE RAZ": PAUSE 100
250 GO TO 170
260 STOP
270
1000 CLS
1010 PRINT "          PAMIETAJ !!! ": PRINT : PRINT
1020 PRINT " Podczas korzystania z miernika ": PRINT "wielozkresowego należy pos
ługi- wać się stałą podziałki 'C'.": PRINT : PRINT
1030 PRINT "            zakres pomiarowy": PRINT "     C=": PLOT 82,100: DRAW 138
,0: PRINT "            liczba działek": PRINT : PRINT : PRINT
1040 PRINT "Aby otrzymać wartość wielkości ": PRINT "mierzonej należy wskazywaną
 li-": PRINT "czbę działek pomnożyć przez sta-": PRINT "łą podziałki(dla aktualn
ie usta-": PRINT "lonego zakresu pomiarowego)."
1050 PRINT #1;" PO PRZECZYTANIU TEKSTU WCIŚNIJ      DOWOLNY KLAWISZ"
1060 PAUSE 0: RETURN
1070
2000 LET POCZ=-48.5
2010 FOR i=0 TO 60 STEP 10: LET xr=125+150*SIN ((pocz+1.625*i)*PI/180): LET yr=1
50*COS ((pocz+1.625*i)*PI/180): PLOT xr,yr
2020 LET xrn=125+160*SIN ((pocz+1.625*i)*PI/180): LET yrn=160*COS ((pocz+1.625*i
)*PI/180): DRAW xrn-xr,yrn-yr: NEXT i
2030 PRINT AT 8,0;"0": PRINT AT 4,4;"5": PRINT AT 1,9;"10": PRINT AT 0,15;"15":
PRINT AT 1,21;"20": PRINT AT 4,26;"25": PRINT AT 8,30;"30"
2040 FOR i=2 TO 60 STEP 2: LET xr=125+150*SIN ((pocz+1.625*i)*PI/180): LET yr=15
0*COS ((pocz+1.625*i)*PI/180): PLOT xr,yr
2050 LET xrn=125+158*SIN ((pocz+1.625*i)*PI/180): LET yrn=158*COS ((pocz+1.625*i
)*PI/180): DRAW xrn-xr,yrn-yr: NEXT i
2060 FOR i=1 TO 60 STEP 2: LET xr=125+150*SIN ((pocz+1.625*i)*PI/180): LET yr=15
0*COS ((pocz+1.625*i)*PI/180): PLOT xr,yr
2070 LET xrn=125+154*SIN ((pocz+1.625*i)*PI/180): LET yrn=154*COS ((pocz+1.625*i
)*PI/180): DRAW xrn-xr,yrn-yr: NEXT i
2080 RETURN
```



FUNKCJA DYSKRYMINUJĄCA

Program „Funkcja dyskryminująca" stanowi narzędzie pomocne w trakcie prowadzenia analizy dyskryminacyjnej dla dwóch P-wymiarowych populacji o liczności prób N1 i N2. Program oblicza podstawowe wielkości potrzebne w trakcie analizy, a mianowicie: współczynniki liniowej funkcji dyskryminującej Fishera, L(P), odległość Mahalanobisa D2, statystykę F, wartości średnie funkcji dyskryminującej Y1SR i Y2SR, wartość funkcji dyskryminującej w kolejnych próbach dla obu populacji Y1 (N1), Y2 (N2) oraz sumaryczną macierz sum i iloczynów A (P, P) z (N1 + N2 — 2) stopniami swobody. Powyższe obliczenia można również przeprowadzić dla zmiennych standardowych (o wariancji równej 1). Możliwa jest wówczas ocena wpływu poszczególnych zmiennych (cech) na wartość funkcji dyskryminującej. Informacje takie wykorzystywane są na przykład w tzw. złożonej metodzie oceny stanów techniki bazującej na analizie dyskryminacyjnej (Devendra Sahal: Foundations of Technometrics, Technological Forecasting and Social Change, Vol. 27, Number 1, February 1985).

Program testowany był w oparciu o dane z badań R. A. Fishera przytoczonych w monografii: M. G. Kendall, A. Stuart: The Advancend Theory of Statistics, Vol. 3, Second Edition (wyd. w jęz. rosyjskim z 1976 r., str. 442).

Dane, X1 (P, N1), X2 (P, N2), wprowadza się instrukcjami DATA poczynając od linii 1300. Kolejność wprowadzania danych (dla P = 2): X1(1, 1), X2 (2, 1),....X1 (1, N1), X1 (2, N1), X2 (1, 1), x2 (2, 1),..., X2 (1, N2), X2 (2, N2). Wprowadzanie danych można przerwać (bez uszczerbku w procesie wprowadzania) w dowolnym momencie, naciskając jednocześnie klawisze **Control i 1,** wznowienie procesu wczytywania następuje po kolejnym naciśnięciu klawiszy.

W celu przyspieszenia działania programu ekran zostaje wyłączony (instrukcja POKE 559, 0) w trakcie obliczeń sum kwadratów i iloczynów (linie 360—430) i wartości zmiennych standaryzowanych (linie 50—180), a te ostatnie obliczenia (szczególnie czasochłonne) umieszczone są w początkowych liniach programu.

```
10 GOTO 1060
20 REM Podprogram:obliczanie zmi
ennych standaryzowanych
30 ? CHR$(125):? "OBLICZANIE F.D
YSKRYMINUJACEJ DLA ZMIENNYCH STA
NDARYZOWANYCH":? :FOR G=1 TO 500
:NEXT G
40 POKE 559,0
50 FOR G=1 TO P:ZMP=0
60 FOR F=1 TO N1:ZMP=ZMP+(X1(G,F
)-X1SR(G))*(X1(G,F)-X1SR(G)):NEX
T F
70 ZMP=ZMP/(N1-1):WAR1(G)=SQR(ZM
P)
80 NEXT G
90 FOR G=1 TO P:FOR F=1 TO N1
100 ZMP=X1(G,F)/WAR1(G):X1(G,F)=
ZMP
110 NEXT F:NEXT G
120 FOR G=1 TO P:ZMP=0
130 FOR F=1 TO N2:ZMP=ZMP+(X2(G,
F)-X2SR(G))*(X2(G,F)-X2SR(G)):NE
XT F
140 ZMP=ZMP/(N2-1):WAR2(G)=SQR(Z
MP)
150 NEXT G
160 FOR G=1 TO P:FOR F=1 TO N2
170 ZMP=X2(G,F)/WAR2(G):X2(G,F)=
ZMP
180 NEXT F:NEXT G
190 OFS=1:POKE 559,34:? CHR$(125
)
200 REM Podprogram: obliczanie w
artosci posrednich
210 REM Obliczanie wartosci X1SR
(P),X2SR(p)
220 FOR G=1 TO P:ZMP=0
230 FOR F=1 TO N1:ZMP=ZMP+X1(G,F
):NEXT F
240 ZMP=ZMP/N1:X1SR(G)=ZMP:? "X1
SR(";G;")=";X1SR(G)
250 NEXT G
260 ? :?
270 FOR G=1 TO P:ZMP=0
280 FOR F=1 TO N2:ZMP=ZMP+X2(G,F
):NEXT F
290 ZMP=ZMP/N2:X2SR(G)=ZMP:? "X2
SR(";G;")=";X2SR(G)
300 NEXT G
310 ? :? "c.d. po nac. ";CHR$(21
0);CHR4(197);CHR$(212):INPUT Q$:
? CHR$(125)
320 REM Obliczanie wektora D
330 FOR G=1 TO P:ZMP=0:ZMP=X1SR(
G)-X2SR(G):D(G)=ZMP:? "D(";G;")=
";D(G):NEXT G
340 ? :? "c.d. po nac. ";CHR$(21
0);CHR$(197);CHR$(212):INPUT Q$:
? CHR$(125):POKE 559,0
350 REM Obliczanie iloczynow mie
szanych
360 FOR G=1 TO P:FOR J=G TO P:ZM
P=0
370 FOR F=1 TO N1:ZMP=ZMP+(X1(G,
F)-X1SR(G))*(X1(J,F)-X1SR(J)):NE
XT F
380 SIX1(G,J)=ZMP
390 NEXT J:NEXT G
400 FOR G=1 TO P:FOR J=G TO P:ZM
P=0
410 FOR F=1 TO N2:ZMP=ZMP+(X2(G,
F)-X2SR(G))*(X2(J,F)-X2SR(J)):NE
XT F
420 SIX2(G,J)=ZMP
430 NEXT J:NEXT G
440 REM Obliczanie iloczynow sre
dnich
450 LI=0:POKE 559,34
460 FOR G=1 TO P:FOR F=G TO P:ZM
P=0:LI=LI+1
470 ZMP=SIX1(G,F)+SIX2(G,F):ZMP=
ZMP/(N1+N2-2)
480 A(G,F)=ZMP:? "A(";G;",";F;")
=";A(G,F)
490 IF G<>F THEN A(F,G)=A(G,F)
500 IF LI=15 THEN LI=0:? "c.d. p
o nac. ";CHR$(210);CHR$(197);CHR
$(212):INPUT Q$
510 NEXT F:NEXT G
520 ? :? "c.d. po nac. ";CHR$(21
0);CHR$(197);CHR$(212):INPUT Q$:
? CHR$(125)
530 REM Podprogram: obliczanie w
spolczynnikow funkcji dyskryminu
jacej (uklad rownan liniowych)
540 FOR G=1 TO P:ZMP=D(G):R(G)=Z
MP:NEXT G
550 ZMP=1.0E-07
560 FOR K=1 TO P:ZMD=ZMP
570 FOR F=K TO P:T=A(F,K):IF ABS
(T)>ABS(ZMD) THEN ZMD=T:G=F:NEXT
 F
580 IF ABS(ZMD)<=ZMP THEN ? "UKL
AD ROWNAN NIE MA JEDNOZNACZNEGO
ROZWIAZANIA":END
590 FOR F=K TO P:T=A(G,F):A(G,F)
=A(K,F):A(K,F)=T:NEXT F:T=R(G):R
(G)=R(K):R(K)=T
600 IF K=P THEN GOTO 630
610 FOR F=K+1 TO P:T=A(F,K)/ZMD:
FOR G=K+1 TO P:A(F,G)=A(F,G)-T*A
(K,G):NEXT G:R(F)=R(F)-T*R(K):NE
XT F
620 NEXT K
630 Z(P)=R(P)/A(P,P)
640 FOR K=P-1 TO 1 STEP -1:Z(K)=
R(K)/A(K,K):FOR G=K+1 TO P:Z(K)=
Z(K)-A(K,G)/A(K,K)*Z(G):NEXT G:N
EXT K

650 REM Wydruk
660 ? CHR$(125)
670 POSITION 2,2:? "WSPOLCZYNNIK
I FUNKCJI DYSKRYMINUJACEJ":? :?
680 FOR F=1 TO P:? "  L";F;"=";Z
(F):NEXT F
690 ? :FOR LI=1 TO 37:? CHR$(160
);:NEXT LI
700 ? :? "c.d. po nac. ";CHR$(21
0);CHR$(197);CHR$(212):INPUT Q$:
? CHR$(125)
710 REM Podprogram: obliczanie w
artosci funkcji dyskryminujacej
720 REM Obliczanie D2
730 ZMP=0
740 FOR G=1 TO P:ZMP=ZMP+Z(G)*D(
G):NEXT G
750 D2=ZMP:? :? :? "D2=";D2
760 REM Obliczanie statystyki
770 ZMP=0:ZMP=(N1*N2*(N1+N2-P-1)
*D2)/(P*(N1+N2)*(N1+N2-2))
780 ? :? "STATYSTYKA=";ZMP
790 REM Obliczanie wartosci sred
nich Y
800 ZMP=0
810 FOR G=1 TO P:ZMP=ZMP+Z(G)*X1
SR(G):NEXT G
820 ? :? "Y1SR=";ZMP
830 ZMP=0
840 FOR G=1 TO P:ZMP=ZMP+Z(G)*X2
SR(G):NEXT G
850 ? "Y2SR=";ZMP
860 ? :? "c.d.po nac. ";CHR$(210
);CHR$(197);CHR$(212):INPUT Q$:?
 CHR$(125)
870 REM Obliczanie wartosci funk
cji dyskryminujacej
880 ? "OBLICZANIE WARTOSCI FUNKC
JI DYSKRYMINU                  JACEJ
":? :FOR ZMP=1 TO 100:NEXT ZMP
890 LI=0
900 FOR F=1 TO N1:ZMP=0
910 FOR G=1 TO P:ZMP=ZMP+Z(G)*X1
(G,F):NEXT G
920 ? "Y1(";F;")=";ZMP
930 LI=LI+1:IF LI=15 THEN LI=0:?
 "c.d. po nac. ";CHR$(210);CHR$(
197);CHR$(212):INPUT Q$
940 NEXT F
950 ? "c.d. po nac. ";CHR$(210);
CHR$(197);CHR$(212):INPUT Q$:? C
```

```
HR$(125)
960 LI=0
970 FOR F=1 TO N2:ZMP=0
980 FOR G=1 TO P:ZMP=ZMP+Z(G)*X2
(G,F):NEXT G
990 ? "Y2(";F;")=";ZMP
1000 LI=LI+1:IF LI=15 THEN LI=0:
? "c.d. po nac. ";CHR$(210);CHR$

(197);CHR$(212):INPUT Q$
1010 NEXT F
1020 ? "c.d. po nac. ";CHR$(210)
;CHR$(197);CHR$(212):INPUT Q$
1030 IF OFS=1 THEN END
1040 GOTO 30
1050 REM Strona tytulowa
1060 ? CHR$(125)
1070 GRAPHICS 2:SETCOLOR 2,0,0:P
OSITION 7,2:? #6;"FUNKCJA":POSIT
ION 4,5:? #6;"DYSKRYMINUJACA"
1080 FOR ZMP=1 TO 200:NEXT ZMP
1090 ? "PODAJ LICZBE ZMIENNYCH P
=";:INPUT P:?
1100 ? "PODAJ LICZBE PROB N1=";:
INPUT N1:?
1110 ? "PODAJ LICZBE PROB N2=";:
INPUT N2
1120 DIM X1(P,N1),X2(P,N2),X1SR(
P),X2SR(P),Q$(3)
1130 DIM A(P,P),SIX1(P,P),SIX2(P
,P),D(P),Z(P),R(P),WAR1(P),WAR2(
P)

1140 GRAPHICS 0:TRAP 1200
1150 REM Wprowadzanie danych
1160 FOR F=1 TO N1:FOR G=1 TO P:
READ L:X1(G,F)=L:? "X1(";G;",";F
;")=";X1(G,F):NEXT G:NEXT F
1170 FOR F=1 TO N2:FOR G=1 TO P:
READ L:X2(G,F)=L:? "X2(";G;",";F
;")=";X2(G,F):NEXT G:NEXT F
1180 ? :? "Koniec wczytywania da
nych":? "c.d. po nac. ";CHR$(210
);CHR$(197);CHR$(212):INPUT Q$:?
 CHR$(125)
1190 OFS=0:GOTO 220
1200 ? "Dane wprowadza sie instr
ukcjami DATA, poczynajac od lini
 1300.";

1210 ? "Kolejnosc wprowadzania d
anych (dla P=2):":? :? "X1(1,1),
X1(2,1),X1(1,2),X1(2,2).....";
1220 ? "X1(1,N1),X1(2,N1); potem
 analogicznie X2(P,N2),gdzie: "
1230 ? :? "X1,X2 - pierwsza i dr
uga grupa danych":? "P - liczba
zmiennych":? "N1,N2 - licznosc p
rob"
1240 ? :? :? "Wprowadz dane!"
1250 STOP
```

```
20000 DATA 7.0,3.2,4.7,1.4
20002 DATA 6.4,3.2,4.5,1.5
20004 DATA 6.9,3.1,4.9,1.5
20006 DATA 5.5,2.3,4.0,1.3
20008 DATA 6.5,2.8,4.6,1.5
20010 DATA 5.7,2.8,4.5,1.3
20012 DATA 6.3,3.3,4.7,1.6
20014 DATA 4.9,2.4,3.3,1.0
20016 DATA 6.6,2.9,4.6,1.3
20018 DATA 5.2,2.7,3.9,1.4
20020 DATA 5.0,2.0,3.5,1.0
20022 DATA 5.9,3.0,4.2,1.5
20024 DATA 6.0,2.2,4.0,1.0
20026 DATA 6.1,2.9,4.7,1.4
20028 DATA 5.6,2.9,3.6,1.3
20030 DATA 6.7,3.1,4.4,1.4
20032 DATA 5.6,3.0,4.5,1.5
20034 DATA 5.8,2.7,4.1,1.0
20036 DATA 6.2,2.2,4.5,1.5
20038 DATA 5.6,2.5,3.9,1.1
20040 DATA 5.9,3.2,4.8,1.8
20042 DATA 6.1,2.8,4.0,1.3
20044 DATA 6.3,2.5,4.9,1.5
20046 DATA 6.1,2.8,4.7,1.2
20048 DATA 6.4,2.9,4.3,1.3
20050 DATA 6.6,3.0,4.4,1.4
20052 DATA 6.8,2.8,4.8,1.4
20054 DATA 6.7,3.0,5.0,1.7
20056 DATA 6.0,2.9,4.5,1.5
20058 DATA 5.7,2.6,3.5,1.0
20060 DATA 5.5,2.4,3.8,1.1
20062 DATA 5.5,2.4,3.7,1.0
20064 DATA 5.8,2.7,3.9,1.2
20066 DATA 6.0,2.7,5.1,1.6
20068 DATA 5.4,3.0,4.5,1.5
20070 DATA 6.0,3.4,4.5,1.6
20072 DATA 6.7,3.1,4.7,1.5
20074 DATA 6.3,2.3,4.4,1.3
20076 DATA 5.6,3.0,4.1,1.3
20078 DATA 5.5,2.5,4.0,1.3
20080 DATA 5.5,2.6,4.4,1.2
20082 DATA 6.1,3.0,4.6,1.4
20084 DATA 5.8,2.6,4.0,1.2
20086 DATA 5.0,2.3,3.3,1.0
20088 DATA 5.6,2.7,4.2,1.3
20090 DATA 5.7,3.0,4.2,1.2
20092 DATA 5.7,2.9,4.2,1.3
20094 DATA 6.2,2.9,4.3,1.3
20096 DATA 5.1,2.5,3.0,1.1
20098 DATA 5.7,2.8,4.1,1.3
```

```
20100 DATA 5.1,3.5,1.4,0.2
20102 DATA 4.9,3.0,1.4,0.2
20104 DATA 4.7,3.2,1.3,0.2
20106 DATA 4.6,3.1,1.5,0.2
20108 DATA 5.0,3.6,1.4,0.2
20110 DATA 5.4,3.9,1.7,0.4
20112 DATA 4.6,3.4,1.4,0.3
20114 DATA 5.0,3.4,1.5,0.2
20116 DATA 4.4,2.9,1.4,0.2
20118 DATA 4.9,3.1,1.5,0.1
20120 DATA 5.4,3.7,1.5,0.2
20122 DATA 4.8,3.4,1.6,0.2
20124 DATA 4.8,3.0,1.4,0.1
20126 DATA 4.3,3.0,1.1,0.1
20128 DATA 5.8,4.0,1.2,0.2
20130 DATA 5.7,4.4,1.5,0.4
20132 DATA 5.4,3.9,1.3,0.4
20134 DATA 5.1,3.5,1.4,0.3
20136 DATA 5.7,3.8,1.7,0.3
20138 DATA 5.1,3.8,1.5,0.3
20140 DATA 5.4,3.4,1.7,0.2
20142 DATA 5.1,3.7,1.5,0.4
20144 DATA 4.6,3.6,1.0,0.2
20146 DATA 5.1,3.3,1.7,0.5
20148 DATA 4.8,3.4,1.9,0.2
20150 DATA 5.0,3.0,1.6,0.2
20152 DATA 5.0,3.4,1.6,0.4
20154 DATA 5.2,3.5,1.5,0.2
20156 DATA 5.2,3.4,1.4,0.2
20158 DATA 4.7,3.2,1.6,0.2
20160 DATA 4.8,3.1,1.6,0.2
20162 DATA 5.4,3.4,1.5,0.4
20164 DATA 5.2,4.1,1.5,0.1
20166 DATA 5.5,4.2,1.4,0.2
20168 DATA 4.9,3.1,1.5,0.2
20170 DATA 5.0,3.2,1.2,0.2
20172 DATA 5.5,3.5,1.3,0.2
20174 DATA 4.9,3.6,1.4,0.1
20176 DATA 4.4,3.0,1.3,0.2
20178 DATA 5.1,3.4,1.5,0.2
20180 DATA 5.0,3.5,1.3,0.3
20182 DATA 4.5,2.3,1.3,0.3
20184 DATA 4.4,3.2,1.3,0.2
20186 DATA 5.0,3.5,1.6,0.6
20188 DATA 5.1,3.8,1.9,0.4
20190 DATA 4.8,3.0,1.4,0.3
20192 DATA 5.1,3.8,1.6,0.2
20194 DATA 4.6,3.2,1.4,0.2
20196 DATA 5.3,3.7,1.5,0.2
20197 DATA 5.0,3.3,1.4,0.2
```

Niespodziewana wizyta

Czytał właśnie gazetę. Gorąca aromatyczna kawa parzyła przyjemnie usta. Najbardziej lubił te chwile, przy porannej prasie. Nieśpieszne, dające jeszcze możliwość błogiego odpoczynku po krótkich nocach spędzonych na szukaniu tematu, rozleniwiające, sycące energią, której tak wiele trzeba do całodziennej pracy. Dziennikarstwo, to paskudnie wyczerpujący zawód.

Na pierwszej stronie, dużymi literami, krzyczał tytuł „Tajemnicze zaginięcie i co dalej? Czy policja wciąż będzie bezradna". Znowu tajemnicza afera. Wiedział, jak nikt inny, że często używa się takich chwytów celowo — dla zdobycia popularności, większej liczby czytelników, zwiększenia nakładu.... Szczególnie, kiedy nadchodził „sezon ogórkowy". A właśnie wakacje, lato w pełni, urlopy...

Dlatego bez większego zainteresowania i z pobłażliwym uśmiechem przebiegł oczyma tekst. Trójkąt bermudzki? I to w pobliżu Phoenix w Arizonie. „Bzdury" — pomyślał. Jeden z jego kolegów po fachu donosił, że wczoraj na autostradzie z Phoenix do Preskott znaleziono kolejny porzucony autobus rejsowy bez pasażerów. Intrygujący był fakt, że po trzydziestoośmioosobowej grupie podróżnych i dwuosobowej obsłudze nie pozostało ani śladu.

Nie znaleziono żadnych bagaży, ani śladów użycia przemocy, jakiejś walki czy paniki. Gdzie podziali się pasażerowie? Poszukiwania na szeroką skalę nie dały żadnych rezultatów. Na te i wiele innych pytań nie znaleziono, jak dotąd, odpowiedzi. Pikanterii dodaje fakt, iż to jest czwarty w tym miesiącu tego rodzaju wypadek na tej właśnie trasie.

Mark odrzucił gazetę na biurko i sięgnął po filiżankę, kiedy zadzwonił telefon. Podniósł słuchawkę. — *Redaktor Mark Henderson?* — Głos mówiącego zdradzał niepokój. — *Nareszcie... Chcieliśmy się z panem skontaktować już wcześniej, ale...*

— *Z kim mam przyjemność?*

— *O, przepraszam, zapomniałem się przedstawić. Doktor Martschewski z kliniki psychiatrycznej w Phoenikshill. Mamy do pana ogromną prośbę... Sprawa jest trochę nietypowa... Oto jeden z naszych pacjentów podaje się za pańskiego stryja...*

— *Co takiego? Mój stryj waszym pacjentem..?*

— *Nie niepokoilibyśmy pana, gdyby nie uporczywe żądania ze strony chorego... Domaga się konfrontacji i potwierdzenia swojej tożsamości przez pana. Proszę nas zrozumieć, przywieziono go bez dokumentów...*

— *Mam tylko jednego stryja... Doktora Percivalla Hendersona...*

— *Takie właśnie wymienił nazwisko...*

— *Ale cóż na miłość boską mógłby robić brat mego ojca w waszej klinice? Z tego co wiem, czuje się wspaniale, jest cenionym chirurgiem... Czyżby ubiegał się o stanowisko konsultanta? — Sytuacja, że doktor Henderson może być w klinice psychiatrycznej, jako pacjent była tak nieprawdopodobna, że próbował żartować.*

— *Sprawa jest bardziej skomplikowana, niż się panu wydaje... Czy mógłby pan przyjechać do nas? Jak najszybciej...*

Odłożył słuchawkę. Przez długą chwilę nie mógł zebrać myśli. Te dziwne niedomówienia doktora Martschewskiego... Klinika psychiatryczna, stryj Percivall, najinteligentniejsza osoba w rodzinie... Co to wszystko ma znaczyć? Wykręcił numer telefonu stryja. Nikt nie podnosił słuchawki. Tyle razy prosił, żeby nie mieszkał sam, w tym niewielkim domku na odludziu...

* * *

Asfaltową aleją podjechał niemal przed samo wejście kliniki. Po kilku zwyczajowych formalnościach przy tego rodzaju sytuacjach pokazano mu pacjenta. Nie mogło być żadnych wątpliwości. To był stryj Percivall. Wkrótce wpuszczono go do niewielkiego pokoju, w którym stało jedynie łóżko trochę dziwnej konstrukcji. Stryj, z rozwichrzonymi włosami, które jakby posiwiały od czasu, kiedy widział go po raz ostatni, przywitał go niecierpliwie, z kwaśnym uśmiechem.

— *A więc jesteś... Nareszcie jesteś... No siadaj, siadaj... Chłopcze, żebyś ty wiedział, co oni tu ze mną wyprawiają...* — To chłopcze brzmiało trochę groteskowo. Mówił przecież do prawie czterdziestoletniego mężczyzny. Było to jego ulubione słowo w rozmowie z Markiem.

— *Ale o co u licha chodzi? Co się stało? Skąd stryj tutaj?*

— *Spokojnie, spokojnie chłopcze. Zaraz ci wszystko opowiem. Tylko... Czekaj, może chcesz się upewnić, czy jestem normalny? No zadawaj pytania...* — Percivall zachowywał się zupełnie normalnie. Nic nie wskazywało, że nagle oszalał. Doktor uprzedzał go jednak, że chory, na pierwszy rzut oka wygląda na normalnego i że pragnie w ten sposób uśpić czujność lekarzy i rodziny.

Mark przysiadł na krawędzi łóżka. Mimo wszystko był nieco sceptycznie nastawiony do dziwnych nastrojów stryja. Czasem nawet miewał wątpliwości... No cóż? Starość nie radość.

— *Wyobraź sobie* — doktor był wyraźnie podniecony — że... *A niech to... Nawet tobie boję się teraz to opowiedzieć... Przecież to właśnie przez to, ja... No wiesz, tutaj przez to trafiłem. To po prostu niesamowita historia... Słuchaj, nie ma chwili do stracenia... Tylko nie myśl, że zwariowałem...*

— *Ależ stryju...*

— *No dobrze już, dobrze... Tylko nie patrz tak na mnie i nie przerywaj, dopóki nie skończę.*

* * *

Za oknem zaszeleściły gałęzie leszczyny rosnącej niedaleko przy nim. Percivall dorzucił kolejną szczapę do kominka. Chciał odpocząć po nurzącym dniu w klinice, przy herbacie, w fotelu. Nie miał już tyle sił, co w młodości. Zdrzemnął się przez chwilę. Widocznie zmarzł trochę, bo przebudził się nagle. Teraz, kiedy płomienie obejmowały suche bierwiono, przyjemne ciepło rozkwitło rumieńcem na jego policzkach.

Usłyszał delikatne stukanie w szybę. Nasłuchiwał przez moment pilnie. Czyżby się przesłyszał? Ale ten odgłos łamanych gałęzi... Nie, jest jak najbardziej realny. Podszedł do okna. Nie mylił się.

— *Niech mnie pan wpuści. Potrzebuję pomocy.* — Przez szybę, w gęstniejącym mroku, niewiele mógł dostrzec. Odniósł jednak wrażenie, że ten człowiek rzeczywiście potrzebuje natychmiastowej pomocy. Zdawało mu się, że w jego oczach, w wyrazie twarzy, wyczytał jakby błagalną prośbę.

— *Niech pan wejdzie, proszę, co się stało?*

W otwartych drzwiach stał mężczyzna średniego wzrostu, mocnej budowy ciała. Z całej jego postaci biła jakaś niezrozumiała groza. Opierał się na sękatej gałęzi, oderwanej niedawno z jakiegoś drzewa. Kikut jego prawej nogi owinięty był w poplamioną twedową marynarkę. Spod pachy drugiej ręki zwisała odcięta, czy

też odrąbana powyżej kolana noga. Mężczyzna wysunął ku niemu rękę z resztką nogi.

— *Niech pan to potrzyma... No, proszę się nie bać. Jest pan przecież chirurgiem, prawda?*

— *Tak, tak... Proszę wejść. Prędzej...* — Profesor usiłował usadzić nieznajomego w fotelu.

— *Już dzwonię do kliniki... Proszę się nie martwić. Uratujemy pana nogę.* — Mówił tak dla dodania otuchy mężczyźnie tak straszliwie doświadczonemu przez los.

— *Niech pan nie dotyka telefonu.* — Głos nieznajomego brzmiał ostro i twardo. Niezwyczajnie, jak na kogoś, kto uległ tragicznemu wypadkowi.

Percivall ani myślał tego słuchać. Był zbyt doświadczonym lekarzem, aby nie wiedzieć, jakie to dziwne rzeczy wygadują ludzie w szoku powypadkowym.

— *Proszę odłożyć słuchawkę. Nigdzie pan nie będzie dzwonił* — usłyszał rozkazujący ton.

„Rzeczywiście jest w szoku" — pomyślał wystukując numer na klawiaturze telefonu. Nagle sękaty kij wytrącił mu słuchawkę z ręki. Trzasnęła głucho o posadzkę. Kostur dusił teraz jego tchawicę.

— *Przyszedłem tutaj, żeby pan mi pomógł, a nie działał na moją szkodę* — mężczyzna akcentował dobitnie każde słowo. — *Szybko — narzędzia... Niech pan przygotuje wszystko do zabiegu.*

* * *

— *Czy ty sobie potrafisz to wyobrazić mój chłopcze?* — Siwa głowa doktora trzęsła się w niespokojnym zdumieniu, jakby raz jeszcze przyszło mu to przeżywać na nowo. — *Sam bym w to wszystko nie uwierzył, gdyby nie... Fakty są najbardziej okrutne.*

Ja do dziś nie mogę zrozumieć, jak to było możliwe... — Odchrząknął, bo poczuł nagle suchość w gardle, która nie pozwalała mu mówić. Z oczu popłynęły łzy.

— *Wyjąłem oczywiście wszystkie narzędzia, jakie miałem w domu. On usiadł na ceratowej kozetce, wiesz tej co stoi w moim ambulatorium i... Drżącymi rękami pomagałem mu wiązać końcówki nerwów, ścięgien, wiązań... Łączyliśmy żyły, naczynia krwionośne, składaliśmy kości.*

Właściwie to on to wszystko robił. Ja mu tylko pomagałem. Rozumiesz... Człowiek bez nogi dokonuje sam operacji... I to jak precyzyjnie... Te jego zręczne palce... A przy tym twarz jego... Dziwnie nieludzka... Ja wiem, że ból zmienia oblicze każdego. Widziałem cierpienie i rozpacz na ludzkich twarzach... On nawet nie zapytał, czy mam coś na miejscowe znieczulenie.

Uwierz mi, taką twarz widziałem po raz pierwszy. Jakaś popielato-biała... Przez cały czas operacji nie drgnął na niej żaden mięsień, ani razu nie wykrzywiła się w grymasie bólu. Wyglądało to wszystko tak, jakby przyszywano nogę manekinowi. To znaczy on przyszywał, bo ja mu tylko asystowałem.

— *Stryju, ale przecież...*

— *A potem stała się rzecz najstraszniejsza.* — Percivall opadł na poduszki. Był tak wzburzony, że wydawało się, iż nie wykrztusi już z siebie ani słowa więcej. „Biedny stryjcio" — pomyślał Mark. Przez tyle lat był dobrym chirurgiem i nagle... Coś mu się w głowie widać jednak poprzestawiało...

— *To oni... Nie odchodź jeszcze...* — wykrzyknął widząc, że Mark podnosi się z łóżka. — *Nie opowiedziałem ci najważniejszego.*

* * *

Nieznajomy mężczyzna wstał z kanapy. Chwiejnym nieco krokiem przeszedł przez pokój.

— *Trzeba jeszcze tylko ukrwić mięśnie i po wszystkim... Dobra robota doktorze.*

— *To przecież niemożliwe!* — Percivall wrzucił narzędzia do sterylizatora. — *To wszystko nie może być realne... Chyba, że pan nie jest...*

— *No proszę, niech pan dokończy doktorze. Chyba, że co...? Że nie jestem człowiekiem, prawda? Tak pan chciał powiedzieć?* — Nieznajomy przywołał na usta coś w rodzaju uśmiechu. — *Ponieważ pomógł mi pan... No dobrze powiem panu prawdę, ale nie wiem, czy to panu wyjdzie na dobre. Czasem lepiej za dużo nie wiedzieć.*

No cóż, skoro pan chce... Na pewnym etapie rozwoju cywilizacji, rozum może nauczyć się kierowania wszystkimi czynnościami biologicznymi każdego organizmu. Przyszycie nogi, ręki, głowy nawet, to wtedy nic nadzwyczajnego.

Doktor patrzył ze zdziwieniem i niedowierzaniem na siedzącego obok człowieka. Człowieka? Czyżby był to... Nie, to jakiś koszmarny sen, z którego zaraz się zbudzi...

— *Nie, nie panie doktorze. Ma pan rację, to nie jest sen.* — Percivall zawisł wzrokiem na nieznajomym. On przecież czytał w jego myślach. — *Wy ludzie nie potraficie sobie wyobrazić ogromu wszechświata, nie chcecie uwierzyć, że mogą istnieć w nim inne, zupełnie odmienne od waszej cywilizacje.*

Czy byłby pan w stanie uwierzyć, że istnieje cywilizacja, która wybrała drogę rozwoju tylko rozumu. I to zupełnie w niewyobrażalnej dla człowieka postaci. Nic, tylko sam rozum.

— *To wszystko... To jest trudne do zrozumienia...* — Doktor odzyskiwał powoli zimną krew. — *Ale dlaczego pan... To znaczy czego wy od nas chcecie? Skąd wzięliście się na Ziemi i po co?*

— *Właśnie chciałem o tym powiedzieć.* — Mężczyzna wstał i zaczął wolno przechadzać się po pokoju. — *Już raz, przed tysiącami lat musieliśmy ingerować w wasz rozwój cywilizacyjny. Wasz potencjał umysłowy rozwijał się zbyt wolno... Teraz czujemy się, jak to wy mówicie, moralnie zobowiązani do opieki nad wami.*

Traktujemy Ziemię, jako takie większe trochę laboratorium. Ziemia jest nam potrzebna i jej mieszkańcy także. Naturalny cykl rozwoju każdej cywilizacji zawsze zmierza ku samozagładzie. Tak było również z naszą planetą. Kiedy doszliśmy do pewnego etapu rozwoju techniki... Wy zbliżacie się właśnie do tego apogeum. Jest już was ponad 5 miliardów. Przy takim tempie prokreacji, jaki macie w Afryce, czy Azji, już wkrótce na każdego mieszkańca Ziemi przypadałoby najwyżej pół metra miejsca. Wy potraficie tylko w jeden sposób rozwiązać ten problem — poprzez wojnę.

Nie macie w ogóle wyobraźni. Czy wiecie, jakie perturbacje wyniknęłyby w Układzie Słonecznym, gdybyście rozsadzili planetę? Przed tysiącami lat nasz rozwój szedł tym samym torem, na który skierowaliśmy was. Niestety, koniec był tragiczny. Została z nas tylko myśl. Czysty rozum. To tylko jest w stanie przetrwać we wszechświecie.

— *Dlaczego w takim razie nie dawaliście znaku życia przez tak długi czas?*

— *Dlaczego? Wy nawet nie jesteście w stanie nas pojąć. Ta ludzka powłoka jest po to właśnie, abyśmy mogli być wśród was. Byśmy mogli wykonać zadanie. Chcemy bowiem, dla waszego dobra, przetransportować część ludzi na inną planetę... Ciekawi pana na jaką, prawda? Na Marsa panie doktorze, na Marsa.*

Jesteśmy tam już od dawna. Nawet narobiliście nam trochę kłopotu w swoim czasie wypatrując przez teleskopy sieć kanałów niezbędną do ładowania naszej energii. Trzeba było szybko je zamaskować, abyście myśleli, że to niewłaściwość przyrządów dała ten efekt.

No i jak pan wie, udało się. Wymyśliliście nawet „efekt oczekiwania", aby wytłumaczyć sobie rozbieżność w obserwacjach. Były kanały i ich nie ma. Pomyłka. Uczeni po prostu tak bardzo chcieli zobaczyć te kanały, że

w końcu je zobaczyli — tłumaczyliście.

— No dobrze, ale do czego potrzebni są wam ludzie tam, na Marsie?

— Są potrzebni... Bardzo potrzebni.

* * *

— Marku, musisz mi pomóc stąd się wydostać — Percivall chwycił go kurczowo za rękę — *No i ratować tych ludzi. Ja byłem już wszędzie. Na policji, u burmistrza... Nikt mi nie uwierzył. Ale ty możesz o tym napisać w prasie. Musisz o tym napisać. Trzeba ostrzec ludzi.* — Drżał teraz na całym ciele, wyczerpany rozmową i strachem.

— Biedny stary — Mark wytarł mu ręcznikiem zroszone czoło. Czy to nie jest przypadkiem rodzinne? — pomyślał nagle. Najpierw babka, teraz...

Drzwi uchyliły się i Martschewski ruchem głowy wywołał go z pokoju.

— Jest w ciężkim stanie — rzekł usprawiedliwiająco. *— Powinien teraz odpocząć. Być może kiedyś jeszcze powróci do zdrowia... Narobił nam wiele kłopotów. Biegał po mieście, zaczepiał ludzi... Na niektórych próbował nawet rzucić się z nożem.*

— Biedny stryj — zgodził się z lekarzem. *— Jeżeli pan pozwoli, będę go czasami odwiedzał. — To taka przykra starość.*

— Tak, nikt z nas nie zna swojego końca — odrzekł lekarz. Uścisnęli sobie ręce i już po chwili Mark pędził autostradą w stronę miasta.

Patrol policyjny zatrzymał go dwadzieścia kilometrów od Phoenix.

— O co chodzi? Chyba nie popełniłem wykroczenia? — Mocne dłonie chwyciły go pod pachy.

— Zaraz się pan wszystkiego dowie.

W asyście trzymających go mocno policjantów przeszedł przez las, gdzie na polanie... Chciał się wyrwać, ale było już na wszystko za późno. Jak zahipnotyzowany patrzył na okolony niebieskawym rozlanym światłem pojazd w kształcie spłaszczonego jaja.

— Więc stryj nie był wariatem, więc to wszystko prawda — wyszeptał drewnianymi wargami, zanim ciśnięto go w otwarty luk.

ANNA KALETA

Przyszłość komputerów

Tygodnik amerykański „PC Week" zamieścił w swoich listopadowych numerach cykl artykułów Jamesa Martina, amerykańskiego producenta sprzętu komputerowego i autora wielu prac z zakresu techniki komputerowej, poświęconych perspektywom rozwoju sprzętu komputerowego w przeciągu najbliższych dwudziestu lat.

Na początku lat dziewięćdziesiątych rozpocznie się międzynarodowy wyścig o dominację w produkcji układów scalonych o submikronowych strukturach. Poprzez zastosowanie ultrafioletowych technik trawienia średnica pojedynczej struktury w układzie scalonym osiągnie wymiar między 0,6 a 0,8 mikrona. Pozwoli to na umieszczenie 100 milionów pojedynczych elementów w układach pamięciowych i 1 miliona w mikroprocesorach.

Dzięki nowym mikroprocesorom średniej klasy komputery będą działać z szybkością 15 milionów operacji na sekundę, wysokiej klasy komputery osobiste z szybkością 100 mln operacji na sekundę, a profesjonalne z szybkością 500 mln operacji na sek., a superkomputery 40 miliardów operacji na sek.

Pojawią się układy scalone, zawierające po kilka procesorów, umożliwiających równoległe przetwarzanie informacji. Powstanie tym samym nowa generacja komputerów tzw. komputerów równoległych.

Tak jak podstawowym osiągnięciem przemysłu komputerowego w latach osiemdziesiątych było pojawienie się komputerów osobistych, tak w latach dziewięćdziesiątych ma nim być masowy rozwój sieci komputerowych umożliwiających powszechne korzystanie z banków informacji i komputerów specjalistycznych.

Sieci te będą budowane ze światłowodów, w których szybkość transmisji będzie wynosić 2,2 miliarda bitów na sekundę.

Coraz powszechniej będą stosowane techniki sztucznej inteligencji umożliwiające nieprofesjonalistom łatwy dostęp do pożądanej informacji. Tworzone będą tzw. systemy ekspertów czyli wysoko specjalistyczne oprogramowania dotyczące poszczególnych dziedzin nauki i techniki, dzięki którym możliwe będzie powszechne korzystanie z najnowszych osiągnięć w tych dziedzinach.

Z końcem lat dziewięćdziesiątych przewiduje się wprowadzenie nowych technik wytwarzania układów scalonych, wykorzystujących promienie rentgenowskie, umożliwiających umieszczenie w jednym układzie ponad 1 miliarda elementów. Pozwoli to na dalsze zwiększenie szybkości pracy komputerów i doskonalenie technik procesów równoległych. Zastosowanie dysków optycznych umożliwi znacznie zwiększenie pojemności pamięci komputerów, dla dysków 4 i 3/4 cala do 8 miliardów bitów, dla 3-calowych do 1 miliarda bitów, a dla 2-calowych do 200 milionów bitów.

W latach dziewięćdziesiątych nastąpi szybki rozwój neuroprocesorów umożliwiających konstruowanie neurokomputerów, czyli komputerów posiadających zdolność samodzielnego uczenia się, wykorzystywanych m.in. do rozpoznawania mowy, rozróżniania i identyfikowania charakterów pisma, analizy rysunków itp. Przewiduje się, że na początku XXI wieku neurokomputery będą zawierały ok. 120 mln neuronów i połączeń między nimi (obecnie do 0,5 mln).

Z początkiem XXI wieku zostanie osiągnięta bariera technologiczna możliwości budowy coraz szybszych mikroprocesorów poprzez zmniejszenie wielkości pojedynczych elementów. Wzrost mocy obliczeniowych komputerów będzie uzależniony przede wszystkim od liczby możliwych połączeń równoległych pracujących w nich procesorów. Pojawią się wówczas komputery szóstej generacji, w których oprócz wielu pracujących równolegle procesorów będą stosowane nadprzewodniki, elementy optyczne, połączone równolegle bazy danych o dużej pojemności na dyskach optycznych, neuroprocesory i kanały optyczne o szybkości transmisji wielu miliardów bitów na sekundę. Z końcem pierwszej dekady XXI wieku najtańsze komputery będą osiągać szybkość działania przekraczającą 80 mln operacji na sekundę, a najdroższe superkomputery 10 bilionów operacji na sekundę, natomiast programy komputerowe niejednokrotnie będą przekraczać 10 mln instrukcji dochodząc nawet do 100 milionów.

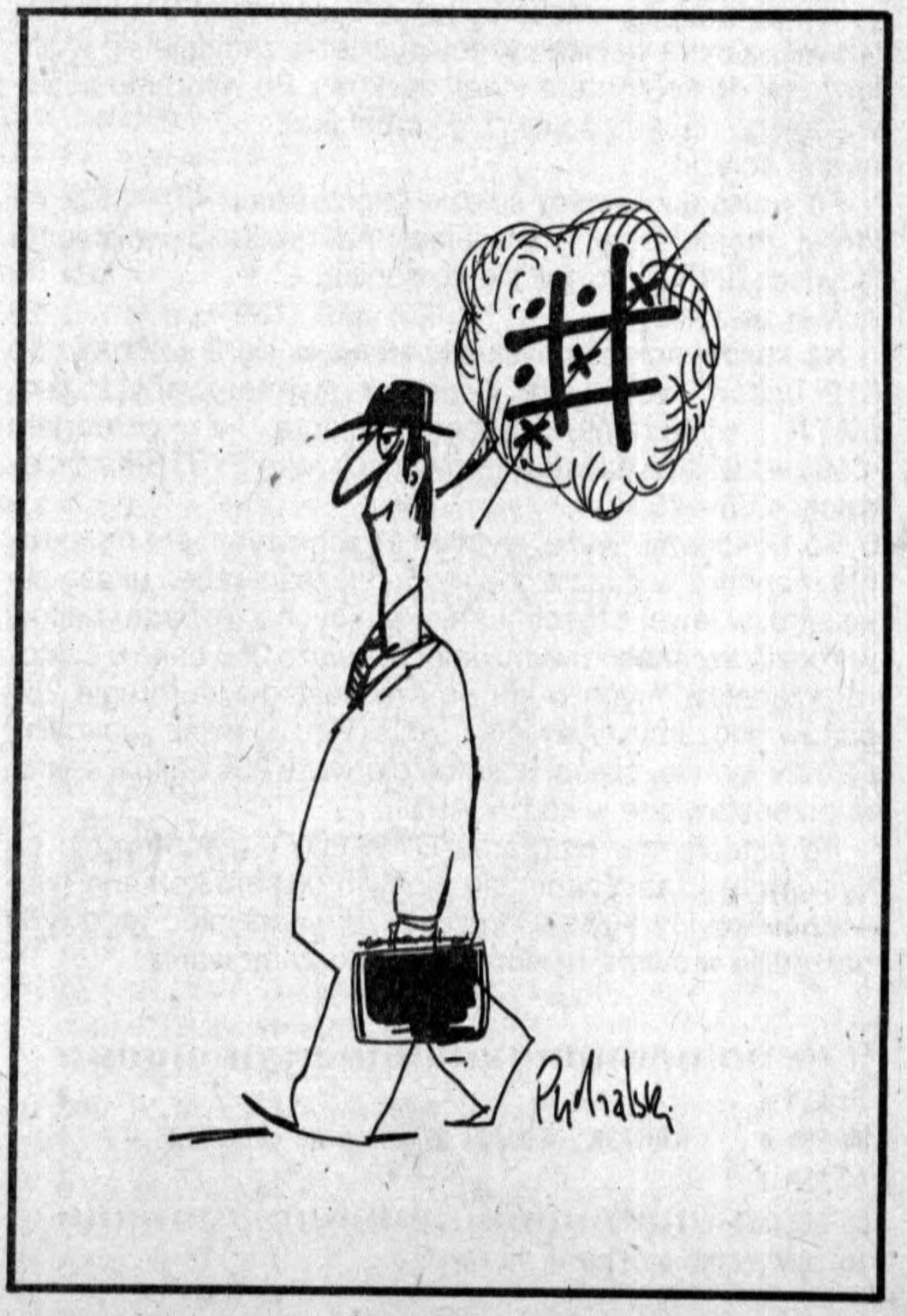
To warto powtórzyć

W podręczniku matematyki dla klasy III szkoły podstawowej znajdują się zadania dotyczące niedziesiątkowych systemów pozycyjnych. Wymagana jest między innymi umiejętność zapisywania liczb naturalnych w systemach o podstawie mniejszej niż 10 oraz zamiany zapisu z innego systemu na zapis w systemie pozycyjnym dziesiątkowym.

```
1 REM Kto zabierze ostatnia litere ?
20 CLS:a=8+INT(RND*9):b=80+INT(RND*120):d=b
30 PRINT"Gra polega na zabieraniu przez graczy - na przemian -
literek x"
40 PRINT"W jednym ruchu mozna zabrac od 1 do ";a;" literek"
50 PRINT"Wygra ten, kto wezmie ostatnia literke"
60 PRINT:PRINT:PRINT:PRINT"s - start"
80 a$=INKEY$
90 IF a$="s" OR a$="S" THEN 110
100 GOTO 80
110 CLS:LOCATE 1,8
120 FOR i=1 TO b:PRINT " x";:NEXT i
130 LOCATE 1,1:PRINT" Kto zaczyna?"
140 PRINT"k - komputer":PRINT"c - czlowiek"
160 a$=INKEY$
170 IF a$<>"k" AND a$<>"K" AND a$<>"c" AND a$<>"C" THEN 160
180 LOCATE 3,20
190 PRINT"Zabieramy od 1 do ";a;" literek"
200 LOCATE 1,1
210 FOR i=1 TO 4
220 PRINT"                                        "
230 NEXT i
240 IF a$="k" OR a$="K" THEN 270
250 IF a$="c" OR a$="C" THEN 390
260 GOTO 130
270 IF b>4*a THEN c=1+INT(RND*a):GOTO 300
280 c=b-(a+1)*INT(b/(a+1))
290 IF c=0 THEN c=1
300 LOCATE 1,1
310 PRINT"                                        "
320 LOCATE 1,1
330 PRINT"Zabieram ";c
340 b=b-c:l=l+c
350 FOR t=1 TO 2000:NEXT t
360 IF b=0 THEN PRINT"Niestety,przegrales":GOTO 500
370 LOCATE 1,8
380 FOR i=1 TO l:PRINT"  ";:NEXT i
390 LOCATE 1,3
400 PRINT"                                        "
410 LOCATE 1,3
420 INPUT"Ile zabierasz ",c
430 IF c<>INT(c) OR c<1 OR c>a THEN 390
440 b=b-c:l=l+c
450 FOR t=1 TO 2000:NEXT t
460 IF b<=0 THEN PRINT"Gratuluje, wygrales !!!":GOTO 500
470 LOCATE 1,8
480 FOR i=1 TO l:PRINT"  ";:NEXT i
490 GOTO 270
500 PRINT"Dowolna litera - nowa gra"
510 a$=INKEY$
520 IF a$="" THEN 510
530 GOTO 10
```

```
10 REM Zamiana liczb naturalnych zapisanych w innych systemach poz
ycyjnych na system dziesiatkowy
20 CLS:DIM t(30)
30 PRINT" Podaj podstawe systemu : 2,3,4,5,6,7,8 lub 9 ";:INPUT a
40 IF a<>INT(a) OR a<2 OR a>9 THEN PRINT"Nie oszukuj":GOTO 30
50 PRINT"Podaj liczbe w systemie o podstawie ";a;" ";:INPUT b
60 d=b
70 IF b<>INT(b) OR b<0 THEN PRINT"Podaj liczbe naturalna":GOTO 50
80 k=k+1:l=10^k
90 IF l>b THEN 110
100 GOTO 80
110 FOR i=k TO 1 STEP -1
120 t(i)=INT(d/(10^(i-1)))
130 IF t(i)>=a THEN PRINT" W tym systemie nie ma takiej liczby, bo
wiem nie ma cyfry ";t(i):GOTO 50
140 d=d-t(i)*10^(i-1)
150 NEXT i
160 PRINT" Podaj zapis tej liczby w systemie dziesiatkowym."
170 PRINT"Jesli sie pomylisz, otrzymasz pomoc ";:INPUT x
180 FOR m=1 TO k
190 y=y+t(m)*a^(m-1)
200 NEXT m
210 IF x=y THEN PRINT"Doskonale":END
220 PRINT"Niestety, nie pamietasz co oznacza zapis ";b;" w systemi
e o podstawie ";a
230 PRINT" Oznacza on, ze jest to suma skladnikow postaci: "
240 PRINT
250 FOR i=k TO 1 STEP -1
260 PRINT t(i);"*";a;"^";i-1
270 NEXT i
280 PRINT:PRINT"Policz teraz ";:INPUT x
290 IF x=y THEN PRINT"Teraz dobrze":END
300 PRINT"Dodaj jeszcze raz":PRINT
310 FOR i=k TO 1 STEP -1
320 PRINT t(i);"*";a^(i-1)
330 NEXT i
340 PRINT:INPUT x
350 IF x=y THEN PRINT"Teraz dobrze":END
360 PRINT:PRINT"Dodaj jeszcze raz":PRINT
370 FOR i=k TO 1 STEP -1
380 PRINT t(i)*a^(i-1)
390 NEXT i
400 PRINT:INPUT x:GOTO 350
```

```
1 REM Zamiana zapisu w systemie dziesiatkowym na zapis w
    innym systemie pozycyjnym
5 CLS
10 PRINT"Wybierz podstawe systemu - 2,3,4,5,6,7,8 lub 9 ";
11 INPUT a
15 IF a<>INT(a) OR a<2 OR a>9 THEN PRINT"Nie oszukuj":GOTO 10
20 PRINT"Wybierz liczbe do zamiany ";:INPUT b
25 IF b<>INT(b) OR b<0 THEN PRINT"Nie oszukuj":GOTO 20
30 d=b:i=-1
35 i=i+1
40 PRINT"Ile to jest ";a;" do potegi ";i;" ";:INPUT c
45 IF c=a^i THEN 55
50 GOSUB 199
51 GOTO 40
55 IF c>b THEN 65
60 GOTO 35
65 DIM t(i)
70 FOR k=i-1 TO 0 STEP -1
75 PRINT"Ile razy ";a^k;" miesci sie w ";d;" ";
76 INPUT t
80 IF t=INT(d/a^k) THEN PRINT "Zapamietaj: ";t:GOTO 90
85 GOSUB 199
86 GOTO 75
90 t(k)=t:f=d:IF k=0 THEN 115
95 PRINT"Ile zostaje ";:INPUT d
100 IF d=f-t*a^k THEN 110
105 GOSUB 199
106 GOTO 95
110 NEXT k
115 FOR m=0 TO i-1:y=y+t(m)*10^m:NEXT m
120 PRINT"Z zapamietanych cyfr utworz liczbe ";
125 INPUT x
130 IF x=y THEN 140
135 GOSUB 199
136 GOTO 125
140 PRINT"To jest liczba ";b;" zapisana w systemie";"o podsta
  wie ";a
145 IF l=0 THEN PRINT "Gratuluje, zadanie rozwiazane bezbled
  nie ":END
150 PRINT"Liczba bledow: ";l:END
199 PRINT"Policz jeszcze raz":l=l+1
200 RETURN
```



Programy narzędziowe na CPC — AMSTRAD 6128

DYSKEDYT

Program DYSKEDYT umożliwia poprawianie dowolnych bajtów każdego sektora wybranej ścieżki dyskietki i jest szczególnie wygodnym narzędziem, gdy chcemy zmienić język polski na języki obcojęzyczne opisy występujące w większości gier komputerowych.

Edycja wybranych bajtów programu (zbioru) odbywa się z poziomu pojedynczego sektora określonej ścieżki dyskietki (interesujący nas sektor jest ściągany do pamięci operacyjnej, a stąd po dokonaniu zmian, zapisywany z powrotem na dyskietkę). Program sygnalizuje każdorazowo błędny odczyt lub zapis sektora. Po wybraniu i wyświetleniu dowolnego sektora możemy wywołać opis dostępnych komend tzw. menu, przyciskając klawisz funkcyjny FO. Z opisem tym trzeba się koniecznie zapoznać, gdyż poniżej piszemy o niektórych tylko komendach używanych w programie.

OPIS DZIAŁANIA PROGRAMU

Po ściągnięciu programu do pamięci i uruchomieniu (komenda RUN „nazwa programu"), w górnym wierszu ekranu pojawi się nazwa programu, dalej wypełniająca prawie cały ekran tabela, a w dolnym wierszu — komunikat: dysk (0,1):. Jeśli korzystamy z napędu umieszczonego w komputerze, to wciskamy kolejno klawisze: "0" i ENTER (klawisz z cyfrą wciskamy, gdy dyskietka znajduje się w dołączonej do komputera stacji dysków). Po wybraniu odpowiedniego napędu pojawi się komunikat:
numer ścieżki:

Po wpisaniu numeru ścieżki (z przedziału <0—39>, na której znajduje się interesujący nas sektor i wciśnięciu klawisza ENTER pojawi się komunikat:
numer sektora:

Na każdej ścieżce dyskietki znajduje się 9 sektorów po 512 bajtów każdy. Dla dyskietek sformatowanych jako DATA wybieramy numer sektora z przedziału <193—201>, a dla sformatowanych jako SYSTEM z przedziału <65—73>.

Po wykonaniu wyżej wymienionych czynności na ekranie monitora w podzielonej na trzy części tabeli ukażą się kolejno: w lewej części numer kolejnych szesnastu bajtów w kodzie szesnastkowym, dalej pierwsze 256 bajtów sektora przedstawionych w kodzie szesnastkowym (drugie 256 bajtów możemy wyświetlić wciskając klawisz funkcyjny F2), a w prawej części te same pierwsze 256 bajtów sektora przedstawione w kodzie ASCII.

Po przepisaniu programu DYSKEDYT (zachowaniu na dyskietce) i zapoznaniu się z całym opisem komend (klawisz funkcyjny F0) stwierdzicie, że warto mieć tego typu narzędzie w swojej biblioteczce oprogramowania.

```
10 REM ************************************************
20 REM *                                              *
30 REM *    AMSTRAD - EDYTOR SEKTOROW NA DYSKIETCE    *
40 REM *                                              *
50 REM ************************************************
60 MODE 2:INK 0,1:INK 1,26:PAPER 0
70 MEMORY &7FFF:GOSUB 1340:' wstawienie kodu maszynowego
80 GOSUB 650:' inicjacja
90 WINDOW SWAP 0,1
100 GOSUB 800:' opis ekranu
110 CLS
120 t=0:cc=0:cr=0:wd=4:wi=24:off=64:v$=d$
130 PRINT CHR$(23);CHR$(1);:TAG #wd:GOSUB 560:POKE &908C,&80:GO
TO 340
140 s$="":f=0:WHILE s$="":c=0:f=f XOR 1:MOVE off+cc*wi+t*8,334-
cr*16:PRINT #wd,CHR$(143);:WHILE s$="" AND c<20:c=c+1:GOSUB 770
:WEND:WEND:IF f=1 THEN MOVE off+cc*wi+t*8,334-cr*16:PRINT #wd,C
HR$(143);
150 address=PEEK(&908B)+PEEK(&908C)*256+cc+cr*16
160 IF ASC(s$)<237 THEN 370
170 IF ASC(s$)<239 THEN GOSUB 1300:IF h$="N" THEN 140
180 IF ASC(s$)=237 THEN k$="zapis sektora":GOSUB 500:GOTO 140
190 IF ASC(s$)=238 THEN 1320
200 IF ASC(s$)<244 THEN 1210
210 IF ASC(s$)=246 THEN 1270
220 IF ASC(s$)=247 THEN GOSUB 940:GOSUB 800:GOSUB 840:GOTO 350
230 cc=0:cr=0:t=0:CLS
240 IF ASC(s$)=245 THEN POKE &908C,PEEK(&908C)XOR 1:GOTO 350
250 IF ASC(s$)=251 THEN sector=sector+1
260 IF ASC(s$)=250 THEN sector=sector-1
270 IF ASC(s$)=248 THEN track=track+1
280 IF ASC(s$)=249 THEN track=track-1
290 IF ASC(s$)=244 THEN GOSUB 560:POKE &908C,&80
300 IF sector>fsn+r THEN sector=fsn:track=track+1
310 IF sector<fsn THEN sector=fsn+r:track=track-1
320 IF track<0 THEN track=39
330 IF track>39 THEN track=0
340 k$="odczyt sektora":GOSUB 500:IF PEEK(&90E3)<>0 THEN 140
350 LOCATE #1,8,3:PRINT#1,drive:LOCATE #1,20,3:PRINT#1,track:LO
CATE #1,32,3:PRINT#1,sector:LOCATE #1,44,3:PRINT#1,PEEK(&908C)A
ND &7F:LOCATE #1,58,3:PRINT#1,a$
360 LOCATE #2,1,1:WINDOW SWAP 0,2:TAGOFF #wd:CLS#3:CLS#4:CLS#5:
CALL &9000:TAG #wd:WINDOW SWAP 0,2:GOTO 140
370 IF t=1 AND wd=4 THEN POKE address,(PEEK(address) AND &F0)+V
AL("&"+s$)
380 IF t=0 AND wd=4 THEN POKE address,(PEEK(address) AND &F)+VA
L("&"+s$)*16
390 IF wd=5 THEN POKE address,ASC(s$)
400 LOCATE #1,9+cc*3,5+cr:k$=HEX$(PEEK(address)):IF LEN(k$)=1 T
HEN PRINT#1,"0";
410 PRINT#1,k$:LOCATE #1,59+cc,5+cr:k$=".":IF PEEK(address)>31
THEN k$=CHR$(PEEK(address))
420 PRINT#1,k$;
430 IF wd=4 THEN t=t XOR 1:IF t=0 THEN cc=cc+1
440 IF wd=5 THEN cc=cc+1
450 IF cc=16 THEN cc=0:cr=cr+1
460 IF cc=-1 THEN cc=15:cr=cr-1
470 IF cr=16 THEN cr=0
480 IF cr=-1 THEN cr=15
490 GOTO 140
500 j=PEEK(&90E5):u=PEEK(&90E6)
510 POKE &90E5,track:POKE &90E4,drive:POKE &90E6,sector:POKE &9
0E3,0
520 IF ASC(k$)=111 THEN CALL &90B1 ELSE CALL &90C3
```

```
530 IF PEEK(&90E3)<>0 THEN CLS:PRINT"BLAD - ";k$;" - sciezka:";
track;" sektor:";sector;CHR$(7);track=j:sector=u:t$="":WHILE t$
=INKEY$:WEND:CLS:RETURN
540 IF ASC(k$)=122 THEN PRINT"prawidlowy zapis sektora";CHR$(7)
550 RETURN
560 GOSUB 840
570 CLS:LOCATE 1,1:PRINT" dysk (0,1): ";:nc=1:mi=0:ma=1:GOSUB 8
60:drive=VAL(w$):LOCATE #1,8,3:PRINT#1,drive
580 IF drive=0 THEN |A ELSE |B
590 fsn=PEEK(PEEK(dph+16*drive)+PEEK(dph+1+16*drive)*256+15)
600 r=8:IF fsn=&41 THEN a$="system" ELSE IF fsn=&C1 THEN a$="da
ta  " ELSE a$="ibm   ":r=7
610 LOCATE #1,58,3:PRINT#1,a$
620 CLS:LOCATE 1,1:PRINT"numer sciezki: ";:nc=2:mi=0:ma=39:GOSU
B 860:track=VAL(w$):LOCATE #1,19,3:PRINT#1,track
630 CLS:LOCATE 1,1:PRINT CHR$(22);CHR$(1);"numer sektora (";CHR
$(8);fsn;CHR$(8);"-";CHR$(8);fsn+r;CHR$(8);"): ";CHR$(22);CHR$(
0);:nc=3:mi=fsn:ma=fsn+r:GOSUB 860:sector=VAL(w$):LOCATE #1,31,
3:PRINT#1,sector;" "
640 CLS:RETURN
650 CALL &908E: 'inicjowanie kodu maszynowego
660 dph=PEEK(&BE40)+PEEK(&BE41)*256+10:d$="0123456789ABCDEF"
670 KEY DEF 14,0,245:KEY DEF 5,0,246
680 KEY DEF 15,0,247:KEY DEF 20,0,237
690 KEY DEF 12,0,238:KEY DEF 13,0,244
700 KEY DEF 2,1,241,249,249
710 KEY DEF 0,1,240,248,248
720 KEY DEF 8,1,242,250,250
730 KEY DEF 1,1,243,251,251
740 POKE &908B,0:POKE &908C,&80
750 WINDOW #1,1,80,22,25:WINDOW #2,2,80,5,25:WINDOW #3,2,6,5,20
:WINDOW #4,9,56,5,20:WINDOW #5,59,75,5,20
760 RETURN
770 s$=INKEY$:IF (s$<CHR$(237) OR s$>CHR$(252)) AND INSTR(""+v$
,UPPER$(s$))<1 AND LEN(v$)>0 THEN s$=""
780 IF v$<>"" THEN s$=UPPER$(s$)
790 RETURN
800 CLS #1:PEN #1,1:LOCATE #1,1,1:PRINT#1,"AMSTRAD - EDYTOR SEK
TOROW NA DYSKIETCE"
810 MOVE 0,380:DRAWR 305,0
820 MOVE 0,72:DRAWR 0,272:DRAWR 600,0:DRAWR 0,-272:DRAWR -600,0
:MOVER 48,0:DRAWR 0,272:MOVER 405,0:DRAWR 0,-272
830 RETURN
840 LOCATE #1,1,3:PRINT#1,"  dysk:   sciezka:     sektor:
strona:      format:       ":RETURN
850 nc=2:mi=2:ma=10
860 w$=""
870 s$="":WHILE s$="" OR LEN(w$)=nc AND s$<>CHR$(127) AND s$<>C
HR$(13) OR s$=CHR$(127) AND w$="":s$=INKEY$:WEND
880 IF s$=CHR$(13) THEN 920
890 IF s$=CHR$(127) THEN w$=LEFT$(w$,LEN(w$)-1):PRINT CHR$(8);C
HR$(16);
900 IF s$<>CHR$(127) THEN PRINT s$;:w$=w$+s$
910 GOTO 870
920 IF VAL(w$)<mi OR VAL(w$)>ma THEN FOR a=1 TO LEN(w$):PRINT C
HR$(8);CHR$(16);:NEXT:PRINT CHR$(7);:GOTO 860
930 RETURN
940 WINDOW SWAP 0,1:CLS
950 REM ********************************************************
960 REM *                                                      *
970 REM *          OPIS CZYNNOSCI EDYCYJNYCH                   *
980 REM *                                                      *
990 REM ********************************************************
1000 LOCATE 31,1:PRINT CHR$(24);" *OPIS KOMEND* ";CHR$(24)
1010 LOCATE 1,3:PRINT CHR$(24);"ustawienie sciezki lub sektora:
";CHR$(24)
1020 PRINT"  <shift> "+CHR$(243)+" ... nastepny sektor."
1030 PRINT"  <shift> "+CHR$(242)+" ... poprzedni sektor."
1040 PRINT"  <shift> "+CHR$(240)+" ... nastepna sciezka."
1050 PRINT"  <shift> "+CHR$(241)+" ... poprzednia sciezka."
1060 LOCATE 1,9:PRINT CHR$(24);" komendy edycji:";CHR$(24)
1070 PRINT"  "+CHR$(243)+" ... nastepny bajt."
1080 PRINT"  "+CHR$(242)+" ... poprzedni bajt."
1090 PRINT"  "+CHR$(241)+" ... nastepna linia."
1100 PRINT"  "+CHR$(240)+" ... poprzednia linia."
1110 LOCATE 1,15:PRINT CHR$(24);" komendy pomocnicze - klawisze
 funkcyjne:";CHR$(24)
1120 PRINT"  f0 ... opis."
1130 PRINT"  f1 ... nowe parametry dysku."
1140 PRINT"  f2 ... numer strony (0 lub 1) sektora."
1150 PRINT"         (0-255 lub 256-511 bajta na strone)."
1160 PRINT"  f3 ... edycja w kodzie hex lub ascii."
1170 PRINT"  f4 ... zapisanie sektora na dysku."
1180 PRINT"  f5 ... wyjscie z programu."
1190 LOCATE 30,24:PRINT CHR$(24);" wyjscie z opisu - SPACJA ";C
HR$(24):s$="":WHILE s$<>" ":s$=INKEY$:WEND
1200 CLS:WINDOW SWAP 0,1:RETURN
1210 IF ASC(s$)=243 THEN 430
1220 IF ASC(s$)=242 AND wd=4 THEN t=t XOR 1:IF t=1 THEN cc=cc-1
1230 IF ASC(s$)=242 AND wd=4 THEN 450
1240 IF ASC(s$)=242 AND wd=5 THEN cc=cc-1:GOTO 450
1250 IF ASC(s$)=240 THEN cr=cr-1:GOTO 470
1260 cr=cr+1:GOTO 470
1270 TAGOFF #wd:wd=wd XOR 1:TAG #wd:t=0
1280 IF wd=4 THEN wi=24:off=64:v$=d$ ELSE wi=8:off=464:v$=""
1290 GOTO 140
1300 IF ASC(s$)=237 THEN k$="zapis sektora " ELSE k$="wyjscie z
 programu "
1310 LOCATE 1,1:PRINT k$;"- JESTES PEWIEN (T/N)?";CHR$(7);:h$="
":WHILE INSTR("TN",h$)<2:h$=UPPER$(INKEY$):WEND:CLS:RETURN
1320 TAGOFF #wd:PRINT CHR$(23);CHR$(0):MODE 1:CALL &BB00
1330 END
1340 IF PEEK(&9000)=&DD THEN RETURN
1350 RESTORE 1430
1360 checksum=0
1370 FOR g=&9000 TO &90DC
1380 READ b$:POKE g,VAL("&"+b$)
1390 checksum=checksum+VAL("&"+b$)
1400 NEXT
1410 IF checksum=30232 THEN RETURN
1420 PRINT"m/c blad danych";CHR$(7)
1430 DATA dd,2a,8b,90,dd,e5,c1,78,e6,7f
1440 DATA 32,8d,90,cd,62,90,79,32,8d,90
1450 DATA cd,62,90,dd,e5,06,03,cd,81,90
1460 DATA 06,10,dd,7e,00,32,8d,90,cd,62
1470 DATA 90,3e,20,cd,5a,bb,dd,23,10,ee
1480 DATA 06,02,cd,81,90,dd,e1,06,10,dd
1490 DATA 7e,00,fe,20,d2,45,90,3e,2e,c5
1500 DATA cd,5d,bb,c1,dd,23,10,ed,3e,0d
1510 DATA cd,5a,bb,3e,0a,cd,5a,bb,dd,e5
1520 DATA c1,3e,00,b9,c2,04,90,c9,21,8d
1530 DATA 90,ed,6f,cd,70,90,ed,6f,cd,70
1540 DATA 90,c9,e6,0f,c6,30,fe,3a,fa,7b
1550 DATA 90,c6,07,c5,cd,5a,bb,c1,c9,c5
1560 DATA 3e,09,cd,5a,bb,c1,10,f7,c9,00
1570 DATA 00,00,dd,21,dd,90,21,db,90,cd
1580 DATA d4,bc,dd,75,00,dd,74,01,dd,71
1590 DATA 02,21,dc,90,cd,d4,bc,dd,75,03
1600 DATA dd,74,04,dd,71,05,c9,21,00,80
1610 DATA ed,5b,e4,90,3a,e6,90,4f,df,dd
1620 DATA 90,d2,d5,90,c9,21,00,80,ed,5b
1630 DATA e4,90,3a,e6,90,4f,df,e0,90,d2
1640 DATA d5,90,c9,3e,ff,32,e3,90,c9,84
1650 DATA 85
```

Koniec „dżumy”

Uczeni radzieccy szukają środka przeciwko niezwykłej chorobie elektronicznej.

W początkach działalności dziennikarskiej miałem szczęście przeprowadzić wywiad z samym Norbertem Winerem — „Ojcem cybernetyki”, jak go wtedy nazywano.

„Człowiek nadaje maszynom cybernetycznym zdolność tworzenia i tym samym stwarza sobie potężnego sojusznika — mówił on — *ale właśnie tutaj czai się niebezpieczeństwo, które może powstać już w najbliższej przyszłości”.*

Winera niepokoiła nasza niezdolność do przekazywania swoich życzeń maszynie w sposób dokładny i wyrazisty. Nie przychodziło mu do głowy, że stworzone z wielkim trudem programy, charakteryzujące się przejrzystością, dokładnością i jednoznacznością, staną się przedmiotem świadomego niszczenia — „terroryzmu” komputerowego. Wątpliwym jest, czy już wtedy uświadamiał on sobie, że już w najbliższych dziesięcioleciach ludzkość uzależni się od komputera.

Pierwsze niepokojące informacje łączono z tajemniczą epidemią, która „poraziła” komputery osobiste, należące do setek tysięcy Amerykanów. Udało się wykryć, że „wirus” został przywieziony z pakistańskiego miasta Lachore, a dokładnie z niedużego sklepu z programami komputerowymi, którego właścicielami byli dwaj bracia Alwi — 26-letni Amdżad Faruk i 19-letni Basit Faruk. Sprzedawali oni dyskietki z zapisanymi na nich „figlarnymi” programami po niezwykle niskich cenach. Naturalnie turyści ulegali pokusie i kupowali je tysiącami, darowując je przyjaciołom i znajomym, nie wiedzieli, że tym samym rozpowszechniają „wirusa” stwarzającego w pamięci maszyny coś na kształt elektronicznego konfetti. Podczas dochodzenia przyznali się, że chcieli „ukarać” Amerykanów. W późniejszym okresie, już we wrześniu br., 40-letni Amerykanin Donald J. Burleson, były programista jednej z firm, specjalnie „zaraził” komputery, żeby zemścić się za jakąś niesprawiedliwość, której w jego mniemaniu firma dopuściła się w stosunku do jego osoby. Wydarzenia potoczyły się lawinowo. Jeden za drugim uszkadzały się komputery, nawet te wykonujące najpoważniejsze operacje. Nawet w Ministerstwie Wojny USA komputery zatrzymywały się lub zmniejszały szybkość operacji, ponieważ ich pamięci wypełnione były bezsensownymi „programami-przybyszami”. I tutaj też stosunkowo szybko znaleziono winowajcę, który o mało co nie stał się przyczyną ogólnonarodowej klęski.

Programy — produkt absolutnie szczególnego rodzaju, zawierają w sobie wielką potęgę intelektualną. Razem z rozwojem techniki komputerowej w tyle pozostały zagadnienia etyki. Zjawisko to stało się problemem natury międzynarodowej.

Swego czasu akademik A. Jerszow, obecny prezes Rady Naukowej „Cybernetyka” Akademii Nauk ZSRR, jeden z największych w świecie teoretyków programowania, opowiadał, jak wysokie mogą być koszty jednego tylko błędu w programie komputerowym.

Pewien programista w Wołżańskiej Fabryce Samochodów ulegający wszelkiego rodzaju zmartwieniom z powodu niedoceniania jego pracy, bawił się uświadamiając sobie swoją władzę nad tokiem procesu produkcyjnego — jak mówił. Znając dobrze program kierujący główną taśmą produkcyjną, doszedł do wniosku, że wystarczy zmienić stan tylko jednego ogniwa w pamięci komputera, a program zacznie zachowywać się czasami normalnie, to znów całkowicie bezsensownie. W pewnym momencie stał się niewolnikiem swojej idei — nie dawała mu ona spokoju. Myśl o takiej możliwości nie odstępowała go. Pewnego razu nieszczęśliwiec wykonał naprawdę szalony zamysł. Załamał się cały system podawania detali na taśmę. Próbowano znaleźć błąd w programie, lecz znajdowano jedynie nieprzewidywane odchylenia. Gdy ostatecznie udowodniono winę programiście, przyznał się on do przestępstwa.

— I co zrobiono z współczesnym Herostratem[1]? — zapytałem.

— Sądzono. Był to u nas pierwszy tego rodzaju proces. Okazało się, że brak w tej sprawie dowodów rzeczowych popełnionego przestępstwa, nie było uszkodzenia sprzętu, niczego nie ukradziono. Występek podsądnego można było zakwalifikować również jako chuligaństwo w produkcji. Ale sprawa nie leży w problemie prawnym, a w tym, że programy muszą posiadać niezawodne zabezpieczenie, znacznie większe niż np. mosty czy obrabiarki. Nie zdajemy sobie sprawy z tego, że uszkodzona konstrukcja np. mostu, będąca skutkiem inżynierskiej pomyłki w obliczeniach, jest nieporównywalnie bardziej kosztowna od ceny (zresztą stosunkowo niewysokiej) dokładnego sprawdzenia programu. Zdarzenie to miało miejsce przed kilkoma laty.

O współczesnej „dżumie” elektronicznej — epidemii, podobnie jak zaraza wirusowa rażącej pamięć komputerową, przerażającej ją w przechowalnię bezsensownych informacji — nikt wtedy nie myślał. Ale uczony potrafił przewidzieć groźne niebezpieczeństwo. Obecnie, gdy informacje o „dżumie komputerowej” grożącej sparaliżowaniem ekonomiki USA wypierają ze szpalt gazet wiadomości poświęcone największym wydarzeniom politycznym, kłopoty Jerszowa stają się już powszechne.

Istota sprawy tkwi w tym, że naprawdę straszny „wirus” poraził nie tylko amerykańskie, ale i radzieckie komputery. Udowodniono tym samym istotną prawdę:

— dzisiejszy świat stał się nierozłączną całością.

Ubiegłego lata w Pierwiesławiu Zalesskim, starym rosyjskim mieście, w którym znajduje się jeden z najmłodszych instytutów akademickich — Instytut Systemów Programowych — odbywało się tradycyjne już spotkanie uczniów radzieckich i zagranicznych, zarażonych „bakcylem” komputerowym. Na letni obóz komputerowy zjechały dzieci z USA, RFN, Włoch, Bułgarii i CSRS. I, prawdopodobnie zupełnie przypadkowo ktoś z gości przywiózł na swojej dyskietce „wirusa”, który został wprowadzony do komputera instytutowego.

Kiedy odnaleziono infekcję komputerową, wydawało się prawdopodobne, że wiele programów zostało zniszczonych i zagłada grozi również pozostałym programom. A przecież koszt oprogramowania wielokrotnie przewyższa koszt najdroższego urządzenia — jakakolwiek maszyna licząca bez programu, to tylko skupisko elektronicznych i mechanicznych urządzeń.

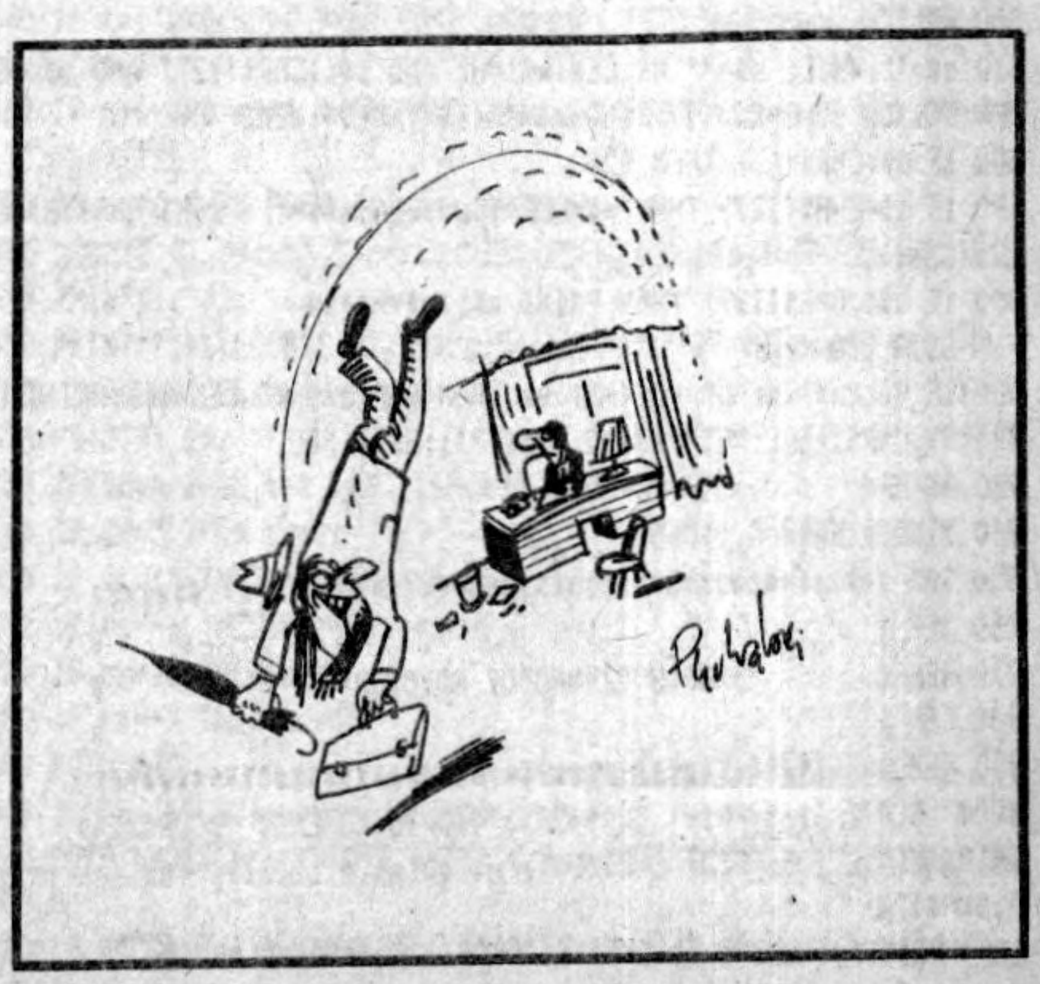
Dyrektor Instytutu Systemów Programowych AN ZSRR, doktor nauk fizyczno-matematycznych, profesor A. Ajłamazjan opowiedział mi o pracy, która umożliwiła nie tylko unieszkodliwienie „wirusa” komputerowego oraz „oczyszczenie” z niego komputerów instytutu, ale również i odtworzenie wszystkich zniszczonych przez niego programów. Jej idea polega na tym, że do „wirusa” komputerowego badacze, na czele których stał kierownik laboratorium systemów programowych dla równoległych architektur S. Abramow, podeszli jak do zwykłego, wirusa białkowego. Krok po kroku, jakby pod mikroskopem, prześledzili „przestępczą” działalność „wirusa”, sprawdzili hipotezy o jego budowie, architekturze jego DNA (kwas dezoksyrybonukleinowy), o możliwym mechanizmie działania „wirusa”, a następnie stworzyli antidotum tj. swego rodzaju program antywirusowy.

Oczywiście jeszcze za wcześnie mówić o tym, że wynaleziono uniwersalny środek walki z „dżumą komputerową”, w obszarze fantastyki pozostają też marzenia o programach — szczepionkach „antywirusowych” dla komputerów. Ale pierwsze kroki zostały uczynione. Udało się nie tylko pokonać jednego konkretnego „wirusa”, ale również wytyczyć kierunki walki z którymkolwiek z nich.

[1]Herostrat — żył w IV w p.n.e.; chcąc zdobyć sławę, podpalił słynną świątynię Artemidy Efeskiej; skazano go na śmierć, a imię jego na zapomnienie; stał się synonimem człowieka zdobywającego rozgłos przez zbrodnię.

Na podstawie artykułu K. Lewitina Koniec „dżumy”? opublikowanego w 317 numerze „Prawdy” z 12.11.1988 r.

Opracowali:
— W. JAGUPOW
— A. KUROCZKIN



INFORMATYCZNY SŁOWNIK ANGIELSKO-POLSKI

I

INTERACTION - wzajemne oddziaływanie, interakcja,
INTERACTION RANGE - obszar wzajemnego oddziaływania,
INTERACTIVE - interakcyjny, dialogowy (np. o systemie, któremu użytkownik wydaje polecenia w trakcie działania programu),
INTERACTIVE DEBUGGER - program umożliwiający poprawianie innych programów poprzez dialog z komputerem,
INTERACTIVE ENVIRONMENT - system konwersacyjny,
INTERACTIVE GRAPHIC - ekranopis przystosowany do dialogu z komputerem,
INTERACTIVE GRAPHICS - grafika interakcyjna (1. organizacja pracy systemu graficznego w którym użytkownik obserwuje i modyfikuje na bieżąco obraz na ekranie monitora, 2. dziedzina programowania związana z opracowywaniem systemów grafiki interakcyjnej),
INTERACTIVE LANGUAGE - język konwersacyjny, język bezpośredniego dostępu,
INTERACTIVELY SELECT - wybór w trybie konwersacyjnym,
INTERACTIVE MODE - tryb konwersacyjny, tryb interakcyjny,
INTERACTIVE OPERATION - praca konwersacyjna, praca interakcyjna,
INTERACTIVE PROCESSING - przetwarzanie konwersacyjne,
INTERACTIVE SYSTEM - system konwersacyjny,
INTERACTIVE UTILITY - konwersacyjny program narzędziowy,
INTERBLOCK GAP - przerwa miedzyblokowa (na nośniku informacji), przerwa blokowa,
INTERCHANGE - 1. wymiana (wzajemna), 2. wymieniać, przestawiać,
INTERCHANGEABLE - zamienny, wymienny,
INTERCHANGEABLY - na przemian,
INTERCHANGE POINT - punkt wymiany,
INTERCOMPUTER COMMUNICATION - komunikacja (łączność) międzykomputerowa,
INTERCONNECT - łączyć wzajemnie,
INTERCONNECTION - wzajemne połączenie, połączenie sprzęgające (sieci),
INTERCONVERSION - przetwarzanie reprezentacji danych,
INTERDEPENDENCE - współzależność, wzajemna zależność,
INTERFACE - 1. interfejs, łącze, sprzęg, złącze standardowe, 2. sprzęgać,
INTERFACE BOARD - tablica złącza (sprzęgu), tablica połączeń interfejsu,
INTERFACE CIRCUIT - układ sprzęgający,
INTERFACE COMPUTER - komputer interfejsowy,
INTERFACE DEVICE - urządzenie złącza (sprzęgu), urządzenie interfejsu (urządzenie zapewniające połączenie mikrokomputera z urządzeniami zewnętrznymi, siecią komputerową lub innymi mikrokomputerami),
INTERFACE-MESSAGE COMPUTER - komputer komunikacyjny,
INTERFACE MESSAGE PROCESSOR - procesor komunikacyjny (służy do przełączania pakietów, marszrut komunikacyjnych, podłączania terminali, utrzymywania łączności z satelitami, szyfrowania informacji, itp.),
INTERFACE MODULE - moduł złącza (sprzęgu), moduł interfejsu,
INTERFACE PROCESSOR - procesor wejścia/wyjścia, procesor sprzęgający,

INTERFACE SPECIFICATION - opis złącza (sprzęgu), opis interfejsu,
INTERFACING - sprzęganie, łączenie,
INTERFERE - wpływać, kolidować,
INTERFERENCE - zakłócenie, interferencja,
INTERFERENCE REJECTION - eliminacja zakłóceń,
INTERFERENCE SUPPRESSION - patrz: INTERFERENCE REJECTION,
INTERFERENCE SUPPRESSOR - eliminator zakłóceń,
INTERFERENCE TRANSMITTER - nadajnik zagłuszający,
INTERFERE WITH ... - kolidować z ...,
INTERFERING FREQUENCY - częstotliwość zakłócająca,
INTERIOR - wnętrze,
INTERLACE - międzyliniowość,
INTERLACED SCANNING - wybieranie międzyliniowe,
INTERLEAVE - przekładać (warstwami), poprzekładać, przeplatać,
INTERLEAVED MEMORY - pamięć o dostępie przeplatanym,
INTERLINE FLICKER - migotanie międzyliniowe,
INTERLISP - odmiana języka Lisp opracowana w firmie Xerox PARC,
INTERLOCK - blokada, uzależnienie, urządzenie ryglujące (programowe lub sprzętowe środki synchronizacji procesów, zapewniające nieprzerwaną pracę w systemach, sieciach, itp.),
INTERLOCKING - blokowanie, uzależnianie, ryglowanie,
INTERMEDIATE - pośredni (np. produkt pośredni, półprodukt),
INTERMEDIATE DATA CARRIER - nośnik danych pośredni,
INTERMEDIATE LANGUAGE - język pośredni (język na który tłumaczony jest program w trakcie pierwszego przebiegu translatora i z którego następuje dalsza translacja),
INTERMEDIATE STORAGE - pamięć pośrednia,
INTERMITTENT - nieciągły, przerywany, nietrwały, przejściowy,
INTERMITTENT ERROR - błąd przejściowy, błąd sporadyczny,
INTERMODULAR REFERENCE - odniesienie międzymodułowe, odwołanie międzymodułowe (np. wykorzystywanie w module nazw zmiennych deklarowanych w innych modułach)
INTERNAL - wewnętrzny,
INTERNAL ABEND - awaryjne zakończenie (np. przetwarzania) spowodowane przyczynami wewnętrznymi,
INTERNAL CHECK - kontrola wewnętrzna,
INTERNAL COMMAND - polecenie rezydentne,
INTERNAL FILE - plik (zbiór) wewnętrzny,
INTERNAL INTERRUPT - przerwanie wewnętrzne,
INTERNAL MEMORY - pamięć wewnętrzna, pamięć operacyjna,
INTERNAL NAME - nazwa wewnętrzna (np. nazwa dostępna tylko w module, w którym jest deklarowana),
INTERNAL PERFORMANCE - szybkość działania procesora (mierzona w liczbie wykonywanych operacji na sekundę),
INTERNAL REFERENCE - odwołanie wewnętrzne, odniesienie wewnętrzne,
INTERNAL REPRESENTATION - przedstawienie wewnętrzne, przedstawienie maszynowe (przedstawienie danych w pamięci komputera, zapewniające łatwość ochrony i przetwarzania),
INTERNAL SCHEMA - schemat wewnętrzny, projekt wewnętrzny,
INTERNAL SORT - sortowanie wewnętrzne,
INTERNAL SPECIFICATION - opis wykonania (opis wewnętrznej struktury programu i sposobu jego działania),
INVERSLY PROPORTIONAL - odwrotnie proporcjonalny,
INVERTED FILE - plik (zbiór) o odwróconym porządku,
INVERTED OUTPUT - wyjście zanegowane,
INVIGILATOR - sygnalizator,
INVISIBLE - niewidzialny, niewidoczny, niedostrzegalny,
INVOCATION - wywołanie (np. podprogramu, procedury),
INVOKE - wywoływać,
IN WORDS - słownie,
IN WRITING - na piśmie,
I/O - patrz: INPUT-OUTPUT,
I/O BOUND TASK - zadanie, którego czas wykonania ogranicza szybkość działania urządzeń wejścia/wyjścia,
I/O CONVERSION - przetwarzanie danych przy wprowadzaniu (z postaci tekstowej na wewnętrzną maszynową) i wyprowadzaniu z komputera (z postaci wewnętrznej na tekstową),
IOCS - patrz: INPUT/OUTPUT CONTROL SYSTEM,
I/O LIMITED PROGRAM - program, którego czas działania ogranicza szybkość działania urządzeń wejścia/wyjścia,
I/O LIST - lista wejścia/wyjścia,
I/O PORT - port wejścia/wyjścia,
IOS - patrz: INPUT/OUTPUT SYSTEM,
I/O WITH HANDSHAKING - przesyłanie z potwierdzeniem i cyklicznym testowaniem stanu urządzenia,
IP - patrz: 1. IMAGE POSITIONING, 2. INSTRUCTION POINTER,
IPL - patrz: INITIAL PROGRAM LOADER,
IRRATIONAL - niewymierny,
IRREGULAR - nieregularny, nieprawidłowy,
IRREVERSIBLE PROCESS - proces nieodwracalny,
IRS - patrz: INFORMATION RETRIEVAL SYSTEM,
IS-A - "jest elementem" (zależność pomiędzy konkretnym obiektem a pojęciem, którego jest on elementem),
ISAM - patrz: INDEXED SEQUENTIAL ACCESS METHOD,
ISARITHMIC CONTROL - sterowanie izorytmiczne,
ISN - patrz: INTERNAL SYSTEM NUMBER,
ISO - patrz: INTERNATIONAL STANDARDS ORGANISATION,
ISOBITS - bity równowartościowe,
ISO CODE - kod ISO (europejski odpowiednik amerykańskiego kodu ASCII),
ISOLATE - wyodrębniać, oddzielać,
ISOLATED WORD - słowo wybrane, słowo wyodrębnione,
ISOLATION - 1. wyodrębnienie, oddzielenie, izolowanie, 2. odłączenie (obwodu),

ISONET - Międzynarodowa Sieć Informacji Normalizacyjnej ISO,
ISR - patrz: 1. INTERRUPT SERVICE ROUTINE, 2. IN SERVICE REGISTER,
ISSUE - 1. wyjście, ujście, wydobywanie się, 2. wynik, rezultat, wydanie (publikacji),
IT - patrz: INFORMATION TECHNOLOGY,
ITEM - 1. pozycja (w spisie), 2. element danych,
ITEM NUMBER - numer pozycji,
ITEM SIZE - 1. rozmiar pozycji, 2. wielkość danych (w bitach, bajtach czy znakach),
ITEM VALUE - wartość (wielkość) elementu danych,
ITERATE - powtarzać, wykonywać iteracje,
ITERATION - iteracja, powtarzanie,
ITERATION STATEMENT - instrukcja cykliczna, instrukcja iteracyjna,
ITERATIVE - powtarzający się,
ITERATIVE ALGORITHM - algorytm iteracyjny,
ITERATIVE PROCESS - proces iteracyjny, proces powtarzający się,
INTERNAL STORE - patrz: INTERNAL MEMORY,
INTERNAL SYSTEM NUMBER - identyfikator obiektu,
INTERNAL TIMER - wbudowany regulator czasowy,
INTERNATIONAL BUSINESS MACHINES CORPORATION - amerykańska korporacja, największy na świecie producent sprzętu komputerowego,
INTERNATIONAL COMPUTATION CENTRE - Międzynarodowy Ośrodek Obliczeniowy,
INTERNATIONAL COMPUTERS LTD. - największy w Wielkiej Brytanii producent sprzętu komputerowego,
INTERNATIONAL FEDERATION FOR INFORMATION PROCESSING Międzynarodowa Federacja Przetwarzania Informacji,
INTERNATIONAL STANDARDS ORGANISATION - Międzynarodowa Organizacja Normalizacji,
INTERNATIONAL TELEGRAPHIC ALPHABET NO 2 - międzynarodowy kod telegraficzny nr 2, kod dalekopisowy CCITT nr 2 (przesyłany znak jest reprezentowany przez 5 bitów),
INTERNET PROTOCOL - protokół wewnątrzsieciowy,
INTERNETTING - współdziałanie międzysieciowe, międzysieciowe oddziaływanie wzajemne,
INTERNETWORKING - współdziałanie międzysieciowe (komunikacja i wzajemne oddziaływanie pomiędzy węzłami różnych sieci komputerowych),
INTERPOLATION - interpolacja,
INTERPOLATOR - kolator, mieszacz,
INTERPRET - tłumaczyć, interpretować, objaśniać,
INTERPRETATION - tłumaczenie, interpretacja,
INTERPRETER - program interpretujący, interpreter, interpretator,
INTERPRETIVE EXECUTION - interpretacja, realizacja (wykonywanie) w trybie interpretacji,
INTERPRETIVE LANGUAGE - język interpretacyjny,
INTERPRETIVE MODE - tryb interpretacyjny,
INTERPRETIVE PROGRAM - patrz: INTERPRETER,
INTERPROCESS COMMUNICATION - współdziałanie procesów,
INTERPROCESSOR INTERFERENCE - interferencja międzyprocesorowa,
INTERRECORD GAP - przerwa między zapisami (rekordami),
INTERRELATION - współzależność, wzajemna zależność, wzajemne powiązanie,
INTERROGATE - zapytywać, zadawać pytania (np. maszynie),
INTERROGATE FEATURE - możność kierowania zapytań do maszyny cyfrowej,
INTERROGATION - zapytywanie, zgłoszenie, odzew,
INTERROGATION PROGRAM - program zapytywania,
INTERRUPT - przerwanie czynności maszyny,
INTERRUPT ACKNOWLEDGE - potwierdzenie przyjęcia przerwania,
INTERRUPT ADDRESS VECTOR - wektor adresów przerwań,
INTERRUPT CHANNEL - kanał przerwań,
INTERRUPT-DRIVEN I/O - układ wejścia/wyjścia sterowany przerwaniami,
INTERRUPT ENABLE - 1. zezwolenie na przerwanie, 2. znacznik włączający lub wyłączający system przerwań,
INTERRUPT ENABLE FLIP-FLOP - przerzutnik zezwalający na przerwania,
INTERRUPT EVENT - zdarzenie powodujące przerwanie,
INTERRUPT FEATURE - możliwość przerywania (np. realizacji programu),
INTERRUPT HANDLER - program obsługi przerwania,
INTERRUPTION - przerwa, przerwanie (np. programu),
INTERRUPT MASK - maska przerwania,
INTERRUPT-MASK REGISTER - rejestr maskowania przerwań,
INTERRUPT PRIORITY - priorytet przerwania,
INTERRUPT REGISTER - rejestr przerwań,
INTERRUPT REQUEST/ACKNOWLEDGE CYCLE - cykl przyjęcia przerwania,
INTERRUPT SERVICE ROUTINE (ISR) - podprogram obsługi przerwań,
INTERRUPT SOFTWARE - 1. oprogramowanie obsługi przerwania, 2. oprogramowanie działające po wywołaniu przerwania,
INTERRUPT STATE - patrz: INTERRUPT STATUS,
INTERRUPT STATUS - stan przerwania,
INTERRUPT SYSTEM - system przerwań,
INTERRUPT TRAP - 1. przerwanie, 2. obsługa przerwania,
INTERRUPT VECTOR - wektor przerwania (jedna lub kilka komórek pamięci, zawierająca adres programu obsługi przerwania i ewentualnie słowo stanu procesora, ustawiane przy wykonywaniu tego programu),
INTERRUPT VECTOR REGISTER - rejestr wektora adresów przerwań,
INTERSECTION - 1. część wspólna (zbiorów), 2. koniunkcja, iloczyn logiczny,
INTERSECTION OF EVENTS - iloczyn zdarzeń,
INTERSEGMENT LINK - połączenie międzysegmentowe,
INTERSEGMENT REFERENCE - odwołanie (odniesienie) międzysegmentowe,
INTERSPACE - odstęp, szczelina,
INTERSYSTEM CROSS-TALK - przesłuch między zespołami,
INTERTASK COMMUNICATION - współdziałanie między zadaniami, wzajemne oddziaływanie zadań,
INTERVAL - przedział czasu, odstęp, interwał,
INTERVAL OF CONVERGENCE - przedział zbieżności,
INTRA-APPLICATION COMMUNICATIONS AREA - obszar komunikacji wewnątrzzastosowaniowej (międzyprogramowej),
INTRICATE - skomplikowany,
INTRINSIC - 1. wewnętrzny, wbudowany, 2. przeznaczony,
INTRINSIC CALL - wywołanie wbudowanej procedury, wywołanie procedury rezydentnej,
INTRINSIC COMMAND - rozkaz rezydentny,
INTRINSIC FUNCTION - funkcja wewnętrzna (nie wymaga definiowania),
INTRINSIC MULTIPROCESSOR - wieloprocesór wewnętrzny,
INTRODUCE - wprowadzać,
INTRODUCE INTO SERVICE - wprowadzić do eksploatacji,
INTRUDER - użytkownik lub program, usiłujący bez zezwolenia dostać się do zbiorów danych,
IN TURN - kolejno, po kolei, w kolejności,
IN USABLE ORDER - w stanie nadającym się do pracy,
IN USE - 1. będący w użyciu, wykorzystywany, 2. zajętość,
INVALID - błędny, nieważny,
INVALID CHARACTER - znak błędny, znak niepoprawny,
INVARIABLE - niezmienny,
INVARIANT - niezmiennik, inwariant,
INVARIANT SYSTEM - układ niezmienny, układ zerozmienny,
INVENTORIES - zapasy (magazynowe),
INVENTORY TAPE - taśma stanów,
INVERSE - 1. odwrotny, 2. odwrotność,
INVERSE IMAGE - przeciwobraz,
INVERSE MATRIX - macierz odwrotna,
INVERSE VIDEO - obraz odwrócony (dot. grafiki),
INVERSION - inwersja, odwrócenie,

J

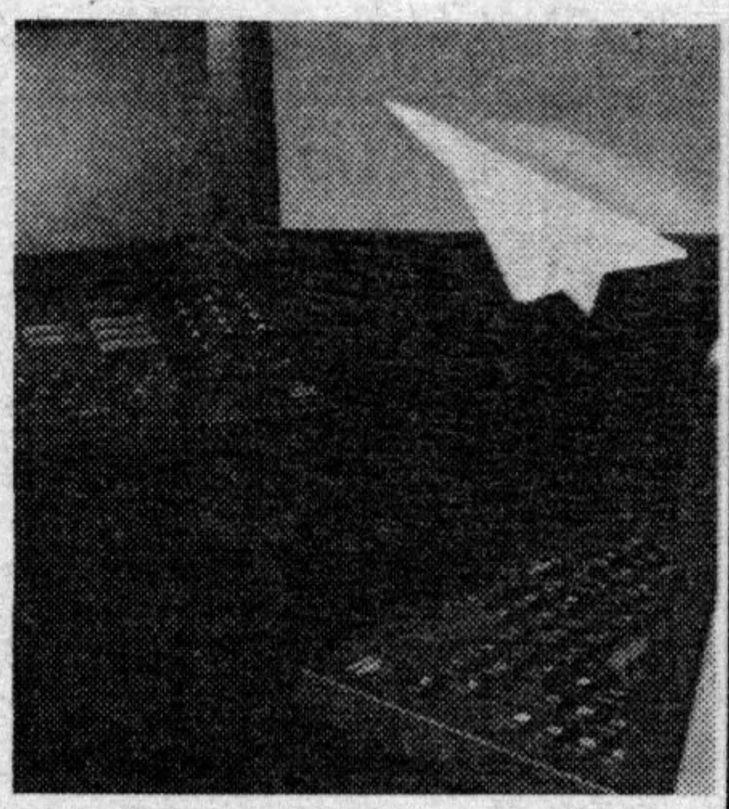
JABBER MONITOR - monitor nadmiernej długości pakietu.
JACK - gniazdko.
JAGGING - 1. nierówność, schodkowaty kształt (czegoś), 2. w grafice rastrowej - zniekształcenie linii z powodu dużego rozmiaru elementów rastra,
JCL - patrz: JOB CONTROL LANGUAGE,
J-K FLIP-FLOP - przerzutnik J-K,
JMP - patrz: JUMP,
JND - patrz: JUST-NOTICEABLE DIFFERENCE,
JOB - 1. zadanie (sekwencja programów), 2. praca (zarobkowa), robota,
JOB ACCOUNTING - rozliczanie zadania,
JOB BATCH - pakiet zadań,
JOB-BLENDING - łączenie w jednej osobie kilku funkcji (np. analityka i programisty),
JOB CONTROL - sterowanie zadaniami (rozdzielenie zasobów komputerowych pomiędzy zadania, ich załadowanie i uruchomienie, zabezpieczenie danych itp.),
JOB-CONTROL CARD - karta sterująca pracy,
JOB CONTROL LANGUAGE - język zarządzania zadaniami, język sterowania pracami,
JOB-CONTROL STATEMENT - dyrektywa,
JOB DECK - pakiet zadań (na kartach dziurkowanych),
JOB DEFINITION - opis zadania, zdefiniowanie zadania,
JOB DESCRIPTION - patrz: JOB DEFINITION,
JOB DESCRIPTION FILE - zbiór opisu zadania,
JOB-DIVIDING - dzielenie zadania do wykonania na części,
JOB FILE - plik (zbiór) zadania (plik, zawierający opis przygotowanego do wykonania zadania),

PIOTRUŚ PAN

Dokończenie ze strony 32

```
 na przesuwaniu wybranego piona
     (FLASH) na wolne pole."
3025 PRINT : PRINT : PRINT TAB 6
;"DOSTEPNE ROZKAZY:"
3030 PRINT : PRINT TAB 6;"<w> -
wybor piona"
3031 PRINT TAB 6;"<r> - ruch pio
na"
3032 PRINT TAB 6;"<i> - informac
ja"
3033 PRINT TAB 6;"<k> - kolory p
lanszy"
3034 PRINT TAB 6;"<n> - nowa gra
"
3100 PRINT #0; INK 9;"     nacisn
ij dowolny klawisz": PAUSE 0: RE
TURN
4000 REM Ustalanie kolorow
4010 INK 9: CLS : PRINT TAB 7; I
NVERSE 1;"USTALANIE KOLOROW"
4020 FOR k=1 TO 4: PRINT AT 2+2*
k,10;n$(k);TAB 19; INVERSE 1; IN
K k(k);"  ": PLOT 151,160-16*k:
DRAW 17,0: DRAW 0,-9: DRAW -17,0
: DRAW 0,9: NEXT k
4030 PRINT : PRINT : PRINT : PRI
NT TAB 5;"wybor elementu <6>,<7>
"
4040 PRINT : PRINT TAB 7;"wybor
koloru <5>,<8>"
4050 PRINT : PRINT TAB 6;"koniec
 wyboru <k>"
4060 LET k=4: LET x$="6": GO TO
4135
4100 LET x$=INKEY$: IF x$="" THE
N GO TO 4100
4110 BEEP .05,20
4120 IF x$="5" THEN LET k(k)=k(k
)-1: LET k(k)=k(k)+8*(k(k)=-1):
PRINT AT 2+2*k,19; INK k(k); INV
ERSE 1;"  "
4125 IF x$="8" THEN LET k(k)=k(k
)+1: LET k(k)=k(k)-8*(k(k)=8): P
RINT AT 2+2*k,19; INK k(k); INVE
RSE 1;"  "
4130 IF x$="7" THEN PRINT AT 2+2
*k,10;n$(k): LET k=k-1: LET k=k+
4*(k=0): PRINT AT 2+2*k,10; INVE
RSE 1;n$(k)
4135 IF x$="5" THEN PRINT AT 2+2
*k,10;n$(k): LET k=k+1: LET k=k-
4*(k=5): PRINT AT 2+2*k,10; INVE
RSE 1;n$(k)
4140 IF x$="k" THEN RETURN
4150 GO TO 4100
9000 REM Pozycje liczb
9010 DATA 4,4
9011 DATA 4,15
9012 DATA 4,26
9013 DATA 11,19
9014 DATA 18,26
9015 DATA 18,15
9016 DATA 18,4
9017 DATA 11,11
9020 REM Polaczenia pol
9030 DATA 2,2,8
9031 DATA 2,1,3
9032 DATA 2,2,4
9033 DATA 3,3,8,5
9034 DATA 2,4,6
9035 DATA 2,5,7
9036 DATA 2,6,8
9037 DATA 3,1,4,7
9040 REM Nazwy elementow
9050 DATA "piony"
9051 DATA " pola"
9052 DATA "linie"
9053 DATA "  tlo"
9100 REM Pozycje pol
9110 DATA 2,2
9111 DATA 2,13
9112 DATA 2,24
9113 DATA 9,9
9114 DATA 9,17
9115 DATA 16,2
9116 DATA 16,13
9117 DATA 16,24
9120 REM Pozycje polaczen
9130 DATA 4,7
9131 DATA 4,18
9132 DATA 18,7
9133 DATA 18,18
9140 DATA 16,24
9141 DATA 16,6
9142 DATA 6,24
9143 DATA 6,6
9990 REM Zapis programu
9995 CLEAR : SAVE "PiotrusPan" L
INE 1: VERIFY "PiotrusPan"
```

Co to jest „Gorąca linia”



Minęło 25 lat od chwili, gdy między Moskwą a Waszyngtonem rozpoczęła pracę tzw. gorąca linia (hot line), czyli linia bezpośredniej łączności między obu stolicami supermocarstw. Nastąpiło to w wyniku porozumienia podpisanego przez Nikitę Chruszczowa i Johna Kennedy'ego, a bezpośrednią przyczyną jej powstania były wydarzenia z jesieni 1962 roku. Wówczas, w trakcie kryzysu karaibskiego świat zbliżył się bardzo niebezpiecznie do nowego konfliktu zbrojnego.

Do tego nie doszło, lecz zdawano sobie sprawę, że takie zagrożenie istnieje, nawet poprzez niedoskonały system wzajemnej informacji dotyczącej posunięć strategicznych obu stron. Rakieta balistyczna lecąca z jednej stolicy do drugiej potrzebowała wtedy 30 minut na przebycie tej drogi. Telefonistka natomiast aż godzinę, aby uzyskać połączenie Moskwa—Waszyngton lub odwrotnie.

W powszechnej opinii gorąca linia kojarzy się z czerwonymi telefonami umieszczonymi w gabinetach prezydenta USA i przywódcy radzieckiego.

Nic podobnego. Od początku istnienia tej linii łączność realizowana była za pomocą dwóch kanałów teleksowych, którymi przekazywano zaszyfrowane depesze.

Praktycznie tak to wygląda do dziś, chociaż technika przekazywania komunikatów została unowocześniona i jest mocno skomplikowana. Początkowo przesyłano depesze za pośrednictwem specjalnego kabla leżącego na dnie oceanu. Potem to łącze zdublowano systemem łączności radiowej na falach krótkich. Stacja przekaźnikowa ulokowana została w Tangerze w Algierii. W 1977 roku, gdy nadeszła era łączności satelitarnej, gorącą linię

Dokończenie na stronie 32

„Papilarny kluczyk”

Ochrona danych dotyczących firmy i prowadzonych przez nią interesów przed konkurencją, to problem wielu biznesmenów posługujących się komputerem. Takie informacje przechowywane są dziś coraz częściej w pamięci tego urządzenia. Aby tę informację uzyskać trzeba znać określone hasło lub kod. Jak uczy praktyka, mogą one zostać rozszyfrowane lub wykradzione.

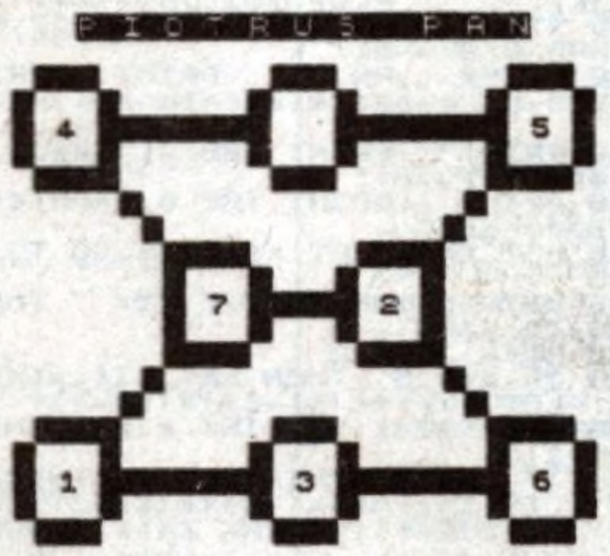
W celu uchronienia się przed taką ewentualnością część firm elektronicznych specjalizuje się w produkcji zabezpieczeń nazywanych biometrycznymi urządzeniami ochronnymi. Są to skomputeryzowane analizatory niepowtarzalnych cech fizycznych człowieka, na przykład głosu czy linii papilarnych. Te urządzenia chronią przed dostępem do nich osób niepowołanych.

Niedawno firma Identix specjalizująca się w produkcji takich zabezpieczeń wypuściła urządzenie przeznaczone dla komputerów osobistych. Dołączone przez złącze RS232C pozwala „nauczyć” komputer rozpoznawania właściciela po odciskach palców. Rozpoznawanie linii papilarnych trwa pierwszy raz kilkanaście sekund, później nie przekracza nawet dwóch sekund. Gdy ktoś nie upoważniony chce skorzystać z komputera, ten nie tylko nie daje się uruchomić, ale też może włączyć system alarmowy.

(Raj)

PIOTRUŚ PAN

```
  1 REM ************************
  2 REM *                      *
  3 REM *              ©1989   *
  4 REM *                      *
  5 REM *  Krzysztof Poźniak   *
  6 REM *                      *
  7 REM *    PIOTRUŚ PAN       *
  8 REM *   gra dla 1 osoby   *
  9 REM *                      *
 10 REM ************************

 20 DEF FN p$(x)=(" " AND x=0)+
(STR$ x AND x<>0)
 100 REM Inicjalizacja programu
 110 CLEAR : INK 7: PAPER 0: INV
ERSE 0: OVER 0: BRIGHT 0: FLASH
0: BORDER 0: CLS
 120 DIM k(4): LET k(1)=7: LET k
(2)=6: LET k(3)=2: LET k(4)=0
 130 RESTORE 9000: DIM a(8,2): F
OR k=1 TO 8: READ a(k,1),a(k,2):
 NEXT k

 140 DIM p(8,4): FOR k=1 TO 8: R
EAD p(k,1): FOR y=1 TO p(k,1): R
EAD p(k,y+1): NEXT y: NEXT k
 150 DIM n$(4,5): FOR k=1 TO 4:
READ n$(k): NEXT k
 160 DIM w(8)
 200 GO SUB 3000: BEEP .1,20
 300 GO SUB 1400
1000 REM Glowna petla programu
1005 GO SUB 2000
1010 PAUSE 0: LET x$=INKEY$: IF
x$="" THEN GO TO 1010
1015 BEEP .05,20
1020 IF x$="w" THEN GO SUB 1500:
 GO TO 1010
1030 IF x$="r" THEN GO SUB 1600:
 GO TO 1010
1040 IF x$="i" THEN GO SUB 3000:
 GO TO 1005
1050 IF x$="n" THEN GO SUB 1300:
 PAUSE 1: GO TO 1010
1060 IF x$="k" THEN GO SUB 4000:

 GO TO 1005
1090 BEEP .1,30: PRINT #0;" ROZK
AZ NIEZROZUMIALY  <i>-inf": PAUS
E 100: LET x$=INKEY$: INPUT "";
IF x$="i" THEN GO SUB 3000: GO T
O 1005
1100 GO TO 1010
1300 REM Nowa gra
1310 PRINT #0," zaczynasz od now
a?   (t/n)"
1320 PAUSE 0: LET x$=INKEY$: INP
UT "": IF INKEY$="t" THEN FOR k=
1 TO 6: BEEP .05,10+5*k: NEXT k:
 GO SUB 1400: GO SUB 2000
1330 RETURN
1400 REM Losowe ustaw. pionow
1410 RANDOMIZE 256*PEEK 23672+PE
EK 23673
1420 LET x$="01234567": FOR k=1
TO 10: LET x=INT (7*RND)+2: LET
x$=x$(2 TO x)+x$(1)+x$(x+1 TO ):
 NEXT k

1430 FOR k=1 TO 8: LET w(k)=VAL
x$(k): IF x$(k)="0" THEN LET p=k
1440 NEXT k: LET r=2: RETURN
1500 REM Przesuniecie wskaznika
1510 PRINT AT a(p(p,r),1),a(p(p,
r),2),w(p(p,r))
1520 LET r=r+1: IF p(p,1)<r-1 TH
EN LET r=2
1530 REM Wyswietlenie wskaznika
1540 PRINT AT a(p(p,r),1),a(p(p,
r),2); FLASH 1;w(p(p,r))
1550 RETURN
1600 REM Przesuniecie pionа /
1610 LET w(p)=w(p(p,r)): LET w(p
(p,r))=0
1620 PRINT AT a(p(p,r),1),a(p(p,
r),2);" "
1630 PRINT AT a(p,1),a(p,2);FN p
$(w(p))
1640 LET p=p(p,r): LET r=2: GO S
UB 1530: RETURN
2000 REM Rysowanie planszy

2010 PAPER k(4): INK k(2): BORDE
R k(4): CLS : PRINT TAB 5; INVER
SE 1;"P I O T R U S   P A N"
2015 RESTORE 9100
2020 FOR k=1 TO 8: READ y,x
2030 PRINT AT y,x+1; INVERSE 1;"
 "; INVERSE 0;TAB x+4; INVERSE 1
;" "
2031 PRINT AT y+1,x; INVERSE 1;"
 "; INVERSE 0;TAB x+4; INVERSE 1
;" "
2032 PRINT AT y+2,x; INVERSE 1;"
 "; INVERSE 0;TAB x+4; INVERSE 1
;" "
2033 PRINT AT y+3,x; INVERSE 1;"
 "; INVERSE 0;TAB x+4; INVERSE 1
;" "
2034 PRINT AT y+4,x+1; INVERSE 1
;"  "
2040 NEXT k
2050 INK k(3): FOR k=1 TO 4: REA
D y,x: PRINT AT y,x; INVERSE 1;"
    ": NEXT k

2060 FOR k=-1 TO 1 STEP 2: FOR l
=-1 TO 1 STEP 2: READ y,x: FOR m
=0 TO 3: PRINT AT y+k*m,x+l*m; I
NVERSE 1;" ";: NEXT m: NEXT l: N
EXT k
2070 PRINT AT 11,14; INVERSE 1;"
 "

2080 INK k(1): FOR k=1 TO 8: PRI
NT AT a(k,1),a(k,2);FN p$(w(k)):
 NEXT k
2090 GO SUB 1530
2100 RETURN
3000 REM Informacja
3010 INK 9: CLS : PRINT TAB 6; F
LASH 1;"I N F O R M A C J A"
3020 PRINT : PRINT : PRINT INVER
SE 1;"PIOTRUS PAN gra dla jednej
 osoby": PRINT : PRINT : PRINT "
 Celem gry jest ulozenie pionow
(cyfr) w/g z gory zalozonej ko-
lejnosci. Drogi ruchu pionow za-
znaczone sa liniami. Ruch polega
```



I N F O R M A C J A

PIOTRUS PAN gra dla jednej osoby

Celem gry jest ulozenie pionow (cyfr) w/g z gory zalozonej kolejnosci. Drogi ruchu pionow zaznaczone sa liniami. Ruch polega na przesuwaniu wybranego piona (FLASH) na wolne pole.

DOSTEPNE ROZKAZY:

<w> - wybor piona
<r> - ruch piona
<i> - informacja
<k> - kolory planszy
<n> - nowa gra

Dokończenie na stronie 31

Co to jest „Gorąca linia”

Dokończenie ze strony 31

poprowadzono przez kosmos. Od czterech lat istnieje też zaszyfrowana łączność teleksowa, dzięki której można przekazywać dokumenty, mapy i inne informacje graficzne.

Na Kremlu obecnie komputerowa końcówka (hot line) składa się z czterech terminali. Dwa z nich — w języku rosyjskim i angielskim — obsługiwane są drogą satelitarną, trzeci podłączony jest do kabla oceanicznego. Czwarty służy do kontaktowania się obsługi technicznej. By utrzymać stałą sprawność systemu gorącej łączności, co godzinę Moskwa i Waszyngton nawiązują łączność. W obu ośrodkach całodobowe dyżury pełnią najwyższej klasy fachowcy — inżynierowie i tłumacze. W godzinach nieparzystych komunikaty kontrolne nadaje strona radziecka, w godzinach parzystych — amerykańska. Są to teksty nie zawierające elementów propagandowych, a najczęściej zawierają fragmenty literatury pięknej lub opisy przyrody. Taki tekst kontrolny musi — w myśl wspólnego porozumienia — być neutralny i zawierać wszystkie litery alfabetu łacińskiego.

W ciągu ćwierć wieku istnienia gorącej linii nie było wypadku, aby łączność się urwała. Współcześnie, nie likwidując starych łączy, a dodając wciąż nowocześniejsze, osiągnięto wielokrotne zdublowanie systemu łączności, które w praktyce uniemożliwia przerwanie kontaktu między Moskwą i Waszyngtonem.

Gorąca linia została uzupełniona specjalnymi ośrodkami zmniejszającymi ryzyko wybuchu konfliktu nuklearnego. Takie porozumienie obie strony zawarły w ubiegłym roku. Ośrodki te, wyposażone w supernowoczesne urządzenia do przekazywania tekstów i obrazów, służą do wymiany niezbędnych informacji dla realizacji układu o likwidacji rakiet jądrowych średniego i krótszego zasięgu, dwóch układów o ograniczeniu podziemnych wybuchów jądrowych, układu o zapobieganiu incydentom na morzu oraz porozumienia o środkach zmniejszania ryzyka wybuchu wojny jądrowej.

(Raj)

„IKS” — dodatek „Żołnierza Wolności”. Redaguje Wiesław Cetera (kierownik zespołu); Rada programowa: Krzysztof Chmarra, Romuald Głąb, Włodzimierz Gogołek, Janusz Janiec, Henryk Krasuski, Ireneusz Miernik, Ludwik Pleia, Jacek Szaniawski. Adres redakcji: 00-950 Warszawa ul. Grzybowska 77, telefon centrali 20-12-61 w. 486. Telex 313664. Rękopisów nie zamówionych redakcja nie zwraca i zastrzega sobie prawo do skrótów. Nakładem: Wydawnictwa „Czasopisma Wojskowe”, Warszawa ul. Grzybowska 77. Fotoskład i druk offsetowy — Wojskowe Zakłady Graficzne im. gen. dyw. A. Zawadzkiego. Nr zam. 146. Nr ind. 361682.

A-52