

P. 1877/82

8-9

1982

---

# informatyka



## W nowy rok

Kończymy rok „wojenny”. Rok nagłego zniechęcenia. Rok wykruszającej się kadry informatyków (często szukających kontraktów za granicą), malejącej komputerowej bazy, rok anemicznego kontaktu między informatykami (piszącymi) a nami. Nowy będzie zapewne spokojniejszy. Osłabły emocje. Zbieramy powoli energię, sumujemy siły, szukamy lepszych dróg na przyszłość.

Nie ma się co oszukiwać: perspektywy nie są dobre. Topniejąca kadra informatyków, nie zwiększająca się liczba komputerów (a więc drastyczny spadek sprzętowego potencjału), niedobory na studiach informatycznych, niechęć użytkowników do wielu dotychczasowych zastosowań, wreszcie: brak zainteresowania informatyką wyraźnie już okazywany przez władze centralne — nie mogą dobrze wróżyć.

Niemniej reforma gospodarcza — wymuszająca efektywność gospodarki (co bez informatyki jest nie do wyobrażenia) — jest już dzisiaj historyczną koniecznością, niezależną od czyjejkolwiek złej lub dobrej woli. Informatyki wyeliminować nie można, chyba że — co byłoby nadto przekornym żartem — sami informatycy postanowiliby ją porzucić.

Nie chcemy pisać, że oto skończył się zły rok, a zaczyna lepszy, bezkonfliktowy. Sami w

to nie wierzymy. Historia ostatnich lat jeszcze bardziej zniechęca do myślenia życzeniowego, opartego na chęci i złudzeniach. Nie ma wątpliwości, że najmniejszy nawet postęp jest następstwem ogromnego wysiłku, determinacji i wiary, że mozolna i niewdzięczna praca przyniesie pożądany efekt i — być może — satysfakcję. Jeśliby więc jednak czegoś życzyć Czytelnikom, to przede wszystkim tego właśnie — determinacji i wiary.

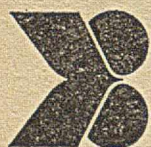
Do niedawna władze państwowe wyraźnie artykułowały własne zapotrzebowanie na informatykę (i ta się dzięki temu rozwijała...), dzisiaj — poza nielicznymi wyjątkami — musi sobie ona radzić sama. Uważamy, że jest to sytuacja korzystna. Informatycy mają bowiem okazję do zrobienia rachunku sumienia i sił, a dzięki temu — do świadomego służenia społeczeństwu.

Zatem jeszcze jedno życzenie na ten nowy, pozbawiony złudzeń rok — skuteczności w polityczkach z biurokratycznymi obwarowaniami i pewnością swego! W rękach społeczeństwa rządzonego racjonalnie — informatyka musi się kiedyś stać głównym jego narzędziem.

REDAKCJA

MACZELNA ORGANIZACJA TECHNICZNA

WYDAWNICTWO



**SIGMA**

ul. Świętokrzyska 14a  
00-950 Warszawa  
skrytka pocztowa 1004

CSASOPISMI / KSIĄŻEK TECHNICZNYCH

### KOLEGIUM REDAKCYJNE

Redaktor naczelny: prof. dr hab. Leon ŁUKASZEWICZ  
prof. dr hab. inż. Konrad FIAŁKOWSKI (zastępca redaktora naczelnego), mgr inż. Zbigniew GLUZA, dr Janusz GWIAZDA, Władysław KLEPACZ (zastępca redaktora naczelnego), dr inż. Tomasz PAWLAK, mgr Andrzej SZALAS, mgr Zbigniew SWIRSKI, dr inż. Janusz ZALEWSKI. Sekretarz redakcji: mgr Teresa JABŁOŃSKA

### RADA PROGRAMOWA

Prof. dr hab. Tadeusz PECHE (przewodniczący), mgr inż. Tomasz BANKOWSKI (sekretarz), mgr inż. Antoni BOSSOWSKI, mgr inż. Roman BURNO, prof. dr hab. Andrzej JANICKI, mgr inż. Jan KRAMARCZUK, prof. dr hab. inż. Juliusz KULIKOWSKI, prof. dr hab. Leon ŁUKASZEWICZ, prof. dr hab. Antoni MAZURKIEWICZ, gen. dr inż. Marian PASTERNAK, dr inż. Bronisław PIWOWAR, mgr Zbigniew SUBSTYK, prof. dr hab. Tadeusz WALCZAK

Materiałów nie zamówionych Redakcja nie zwraca.

Redakcja: 00-041 Warszawa, ul. Jasna 14/16, pok. 131 i 133, tel. 27-71-40, dyżury redakcji 10.00—12.00

Zakł. Graf. „Tamka”. Zam. 294. Obj. 5,5 ark. druk. Nakład 5000 egz. M-97.

Cena egzemplarza zł 50.—

INDEKS 36124

Prenumerata roczna zł 600.—



P. 1897/82  
Biuletyn  
Lubuski  
Techniczny

Kreczmar A., Salwiński A.: Język programowania LOGLAN. Część 2

INFORMATYKA 1982, nr 8-9, s. 4

Druga część prezentacji języka LOGLAN zawierająca omówienie dalszych elementów jego konstrukcji: tablic dynamicznych, klas, prefiksowania, współprogramów oraz typów formalnych.

Кречмар А., Сальвицкий А.: Язык программирования LOGLAN. Часть 2

INFORMATYKA 1982, № 8-9 стр. 4

Вторая часть представления языка LOGLAN, содержащая обсуждение дальнейших элементов его конструкции: динамических досок, классов, префиксирования, сопрограмм и формальных типов.

Dawidowski J., Owczarczak P., Wiśniewski M.: Dwie metody badania sprawności systemów komputerowych.

INFORMATYKA 1982, nr 8-9, s. 7

Prezentacja dwóch różnych metod badania sprawności systemów komputerowych, ilustrowana wynikami badań sprawności systemu WASK eksploatowanego w poznańskiej Akademii Ekonomicznej.

Давидовски И., Овчарчак П., Висьневски М.: Два метода исследования исправности вычислительных систем

INFORMATYKA 1982, № 8-9 стр. 7

Представление двух разных методов исследования исправности вычислительных систем, поясняемое результатами исследования точности системы эксплуатированной в Познаньской Экономической Академии.

Biskupski P.: Analiza sygnatur — uniwersalna metoda wykrywania uszkodzeń w układach cyfrowych

INFORMATYKA 1982, nr 8-9, s. 11

Charakterystyka metody wprowadzonej przez firmę HEWLETT PACKARD. Metoda ta w porównaniu do metod poprzednio stosowanych pozwala znacznie zwiększyć skuteczność wykrywania uszkodzeń.

Бискупски П.: Анализ шифров — универсальный метод обнаруживания неисправностей в цифровых устройствах

INFORMATYKA 1982, № 8-9 стр. 11

Характеристика метода внедренного фирмой HEWLETT-PACKARD. Метод этот, по сравнению с методами прежде применяемыми, позволяет значительно увеличить эффективность обнаруживания неисправностей.

Sobczyk M.: Algorytmy sortowania bez porównań

INFORMATYKA 1982, nr 8-9, s. 14

Charakterystyka jednej z klas algorytmów sortowania. Podano podstawowe określenia oraz omówiono poszczególne rodzaje algorytmów tej klasy, ilustrując je danymi porównawczymi szybkości działania różnych algorytmów.

Собчык М.: Алгоритмы сортировки без сравнений

INFORMATYKA 1982, № 8-9 стр. 14

Характеристика одного из классов алгоритмов сортировки. Представлено основное определение и обсуждено отдельные роды алгоритмов этого класса, иллюстрируя их сравнительными данными скорости функционирования разных алгоритмов.

Kukuła A., Szmidt U.: Analiza przydatności USZBD RODAN dla baz danych o wysokiej dyspozycyjności

INFORMATYKA 1982, nr 8-9, s. 16

Omówienie wyników badań nad przydatnością systemu RODAN do zarządzania bazami danych w systemie informatycznym huty żelaza. Przeprowadzono analizę narzędzi systemu RODAN do ochrony danych oraz dokonano porównania aktualnych możliwości funkcjonalnych tego systemu z innymi systemami zarządzania bazami danych.

Кукула А., Шмидт У.: Анализ пригодности USZBD RODAN для баз данных с высокой распоряженностью

INFORMATYKA 1982, № 8-9 стр. 16

Обсуждение результатов исследований на полезности системы RODAN для управления базами данных в системе металлургического завода. Проведено анализ инструментов системы RODAN для защиты данных и совершено сравнение актуальных функциональных возможностей этой системы с другими системами управления базами данных.



Kreczmar A., Salwicki A.: LOGLAN programming language. Part 2

INFORMATYKA 1982, No. 8-9, p. 4

Second part of the LOGLAN language presentation, which include discussion on further elements of their construction: dynamic tables, classes, prefixing, coprograms and formal types.

Kreczmar A., Salwicki A.: LOGLAN Programmiersprache. Teil 2

INFORMATYKA 1982, Nr. 8-9, S. 4

Zweiter Teil der LOGLAN-Sprache Präsentation, die weitere Elemente ihrer Bauweise umfasst: dynamische Tafeln, Klassen, Prefixierung, Mitprogramme und formelle Typen.

Dawidowski J., Owczarczak P., Wiśniewski M.: Two methods for computer system efficiency research

INFORMATYKA 1982, No. 8-9, p. 7

Presentation of two different methods for computer system efficiency research, illustrated with research results on WASK system in the Poznań Economic Academy.

Dawidowski J., Owczarczak P., Wiśniewski M.: Zwei Methoden zur Forschung der Rechnersystemeleistung

INFORMATYKA 1982, Nr. 8-9, S. 7

Eine Vorstellung von zwei verschiedenen Methoden zur Forschung der Rechnersystemeleistung, die mit Forschungsergebnissen über WASK-System in der Ökonomischen Akademie Poznań illustriert wurden.

Biskupski P.: Signature analysis — a universal method for error detection in digital circuits

INFORMATYKA 1982, No. 8-9, p. 11

Characteristics of the by HEWLETT-PACKARD introduced method. The method, in comparison to previously used methods, makes it possible to extend the efficiency of error detection.

Biskupski P.: Signaturenanalyse — eine universale Methode der Fehlererkennung in Digitalschaltungen

INFORMATYKA 1982, Nr. 8-9, S. 11

Eine Charakteristik der von der Firma HEWLETT-PACKARD eingeführten Methode. Im Vergleich zu den früher verwendeten Methoden erlaubt diese Methode die Wirksamkeit der Fehlererkennung stark zu steigern.

Sobczyk M.: Algorithms for sorting without comparisons

INFORMATYKA 1982, No. 8-9, p. 14

Characteristics of one class of sorting algorithms. Basic terms are presented, as well as individual algorithms of this class are discussed, illustrated with comparable data of different algorithms operating speed.

Sobczyk M.: Algorithmen für Sortierung ohne Vergleichen

INFORMATYKA 1982, Nr. 8-9, S. 14

Eine Charakteristik einer von Sortieralgorithmeklassen. Es wurden die Grundbegriffe angegeben, sowie die einzelnen Algorithmen dieser Klasse, illustriert mit Vergleichsdaten über Operationsgeschwindigkeit der verschiedenen Algorithmen, besprochen.

Kukuła A., Szmidi U.: Usability analysis of the data base management system RODAN for data bases with high readiness

INFORMATYKA 1982, No. 8-9, p. 16

Research results on the RODAN system usability for data bases management in steelworks data processing system are discussed. The analysis of the RODAN system tools for data protection is made and actual functional possibilities of the system in relation to other data base management systems are compared.

Kukuła A., Szmidi U.: Nützlichkeitsanalyse des Datenbasisverwaltungssystems RODAN für Datenbasen mit hoher Verfügbarkeit

INFORMATYKA 1982, Nr. 8-9, S. 16

Eine Besprechung der Forschungsergebnisse hinsichtlich Verfügbarkeit des RODAN-Systems für Verwaltung von Datenbasen in EDV-System eines Eisenhüttenwerkes. Es wurden Hilfsmittel des RODAN-Systems für Datenschutz analysiert, sowie jetzige Funktionsmöglichkeiten dieses Systems mit Möglichkeiten der anderen Datenbasisverwaltungssysteme verglichen.



---

**ORGAN KOMITETU INFORMATYKI MINISTERSTWA NAUKI, SZKOLNICTWA WYŻSZEGO  
I TECHNIKI ORAZ KOMITETU NAUKOWO-TECHNICZNEGO NOT DS. INFORMATYKI**

---

<b>W NUMERZE:</b>	<b>Strona</b>
Spis treści rocznika 1982	wkładka
Język programowania LOGLAN. Część 2 <i>Antoni Kreczmar, Anarzej Salwicki</i>	4
Dwie metody badania sprawności systemów komputerowych <i>Jan Dawidowski, Piotr Owczarczak, Marek Wiśniewski</i>	7
Analiza sygnatur — uniwersalna metoda wykrywania uszkodzeń w układach cyfrowych <i>Paweł Biskupski</i>	11
Algorytm sortowania bez porównań <i>Marek Sobczyk</i>	14
<b>SYSTEMY</b>	
Analiza przydatności USZBD RODAN dla baz danych o wysokiej dyspozycyjności <i>Andrzej Kukuła, Urszula Szmidt</i>	16
Programowany rejestrator PRS-4 — podstawa automatyzacji kopalni węgla kamiennego <i>Marek Mokrosz, Janusz Suchy, Krystian Żymetka</i>	19
Współdziałanie resortowego systemu informatycznego MHiPM z innymi krajowymi systemami handlu zagranicznego <i>Leon Kasman</i>	22
AESN — pakiet programów do generowania wersji nakładanych programów fortranowych dla ODRY 1305 <i>Leszek Rakowski</i>	27
<b>ALGORYTMY</b>	
Sortowanie przez podział dystrybucyjny <i>Marek Sobczyk, Andrzej Szalas, Zbigniew Swirski</i>	29
O pewnym rozwiązaniu układu równań liniowych <i>Danuta Kalinowska</i>	31
<b>Z KRAJU</b>	
Zastosowania komputerów w przemyśle <i>Janusz Zalewski</i>	32
Kadra kierownicza PGR o potrzebie zastosowań informatyki <i>Jadwiga Orylska</i>	34
Narzędzie ekonomisty? (J.G.)	35
GIEŁDA INFORMACJI	33—35
<b>ZE ŚWIATA</b>	
DATASHOW'82 <i>Oprac. Marianna Sobczyk</i>	36
Informatyka we Francji — kilka liczb <i>Oprac. Danuta Segiet</i>	36
<b>TERMINOLOGIA</b>	
O testowaniu, weryfikacji i uwiarygodnieniu programów <i>Janusz Zalewski</i>	38
<b>POGLĄDY</b>	
Selekcja negatywna <i>Janusz Gwiazda</i>	III okł.
Ot, marzenie <i>Zbigniew Gluza</i>	III okł.



## Język programowania LOGLAN. Część 2

W poniższej, drugiej części cyklu artykułów o LOGLANIE zaprezentowane są dalsze (po omówionych w części pierwszej — w poprzednim numerze) konstrukcje występujące w tym języku. Ostatnia, trzecia część zawierać będzie opis koncepcji programowania współbieżnego, obsługi sygnałów, gospodarki pamięcią oraz oddzielnej kompilacji modułów.

### TABLICE DYNAMICZNE

W języku LOGLAN tablice są dynamiczne. Tablice dynamiczne należy najpierw zadeklarować, a następnie wygenerować w ciągu instrukcji modułu. Rozważmy najprostszy przykład:

```
block
  var n, i: integer;
  var A: arrayof integer; (* tutaj jest deklaracja tablicy A *)
begin
  read(n);
  array A dim (1:n); (* tutaj jest generacja tablicy A *)
  for i:=1 to n do read (A(i)) od;
  (* itd. *)
end
```

Mamy tu zadeklarowaną zmienną tablicową  $A$ . Wartością jej może być adres tablicy elementów typu *integer*. Instrukcja generacji tablicy:

```
array A dim (1:n);
```

powoduje wygenerowanie odpowiedniego pola na tablicę jednowymiarową o zakresie indeksu od 1 do  $n$ , a następnie przekazanie adresu tego pola na zmienną  $A$ .

Ogólnie, syntaktyka takiej generacji jest następująca:

```
array A dim (lower:upper)
```

gdzie *lower* i *upper* są dowolnymi wyrażeniami arytmetycznymi definiującymi odpowiednio dolny i górny zakres przebiegu wskaźnika.

Przykładowa deklaracja tablicy dwuwskaźnikowej może być następująca:

```
var A: arrayof arrayof integer;
```

Aby wygenerować taką tablicę, należy najpierw wygenerować jeden wiersz:

```
array A dim (1:n);
```

a następnie dla każdej zmiennej  $A(i)$  wygenerować odpowiednią kolumnę:

```
for i:=1 to n do array A (i) dim (1:m) od;
```

Po tych wstępnych informacjach rozważmy następujący przykład:

```
block
  var i, j: integer, A, B: arrayof arrayof real, n: integer;
begin
  read (n);
  array A dim (1:n);
  for i:=1 to n do array A (i) dim (1:n) od;
  (* A jest tablicą kwadratową *)
  array B dim (1:n);
  for i:=1 to n do array B (i) dim (1:i) od;
  (* B jest tablicą trójkątną dolną *)
  A(n, n) := B(n, n);
  B(1) := A(1);
  B(1) := copy (A(1));
end
```

Generujemy tu najpierw tablicę kwadratową  $A$ , o wymiarach  $n$  na  $n$ . Każdy element  $A(i)$ ,  $1 \leq i \leq n$ , zawiera adres wiersza o długości  $n$ . Tablica  $B$  ma natomiast inną strukturę. Każdy element  $B(i)$ ,  $1 \leq i \leq n$ , zawiera adres wiersza o długości  $i$ .

Zatem instrukcja przypisania  $A(n, n) := B(n, n)$  powoduje przesyłanie odpowiedniej wartości rzeczywistej z  $B(n, n)$  na  $A(n, n)$ . Natomiast instrukcja przypisania  $B(1) := A(1)$  powoduje przesyłanie adresu pierwszego wiersza tablicy  $A$  na zmienną  $B(1)$ . Wreszcie instrukcja przypisania  $B(1) := \text{copy}(A(1))$  powoduje utworzenie kopii pierwszego wiersza tablicy  $A$  i przesyłanie adresu tej kopii na zmienną  $B(1)$ .

Wartości dolnego i górnego zakresu wskaźnika tablicy są dostępne poprzez operatory *lower* i *upper*, których argumentem może być dowolna tablica dynamiczna.

Należy jeszcze zaznaczyć, że wartość zmiennej tablicowej może być nieokreślona (np. przed wykonaniem pierwszej generacji podstawiającej na nią adres odpowiedniej tablicy). Wówczas wartość takiej zmiennej jest równa *none* (standardowe określenie nieistniejącego pola). Próba dostępu do elementu takiej tablicy, np.  $A(i)$ , lub wyznaczenia zakresu wskaźnika, np. *lower(A)*, będzie wykryta w czasie wykonywania programu i potraktowana jako błąd.

### KLASY

Klasy są istotnym uogólnieniem pojęcia typu strukturalnego (rekordu). Przypuśćmy, że chcemy w programie zdefiniować złożony typ danych. Możemy go zbudować z typów pierwotnych: *integer*, *real*, *boolean*, *char* (i ewentualnie innych).

Rozpocznijmy od przykładu. Poniższa deklaracja:

```
unit stos: class (n: integer);
  var top: integer, A: arrayof integer;
  unit pop: function: integer;
begin
  if top > 0 then result := A(top); top := top-1 fi
end pop;
unit push: procedure (x: integer); (* x — element wstawiany *)
begin (* uwaga, jeżeli stos się przepelnił, wywołujemy procedurę alarm *)
  if n > top then top := top+1; A(top) := x
  else call alarm
end
```



```

fi
end push;
begin
array A dim (1:n);
end stos;

```

jest deklaracją klasy, która ma składniki (zwane atrybutami) w kolejności: *top (integer)*, *A(array of integer)*, *pop (function)*, *push (procedure)*. Parametrem tej klasy jest liczba całkowita *n*. Sposoby transmisji parametrów i ich rodzaje są takie same jak dla procedur.

W klasie mogą być zdefiniowane pewne akcje inicjujące. Wówczas z chwilą generacji obiektu klasy, po przesłaniu parametrów wejściowych, sterowanie przechodzi do ciągu instrukcji klasy (tak jak ma to miejsce w procedurach). Powrót do miejsca generacji wykonuje się tak samo jak powrót do miejsca wywołania procedury, czyli po natrafieniu na *return* lub końcowe *end* w klasie. Dla procedur lub funkcji powrót taki łączy się z zakończeniem działania modułu i usunięciem jego pola z pamięci komputera. Dla klas, po powrocie do miejsca wygenerowania, nie usuwa się takiego pola i jego adres może być przesyłany na zmienną referencyjną (np. po to, aby można było operować na tym polu za pomocą zdalnego dostępu). Zmienne referencyjne (podobnie jak zmienne tablicowe) mogą mieć wartość *none*.

Założmy, że mamy w naszym programie deklaracje zmiennych typu *stos*, np:

```

var s, t, z:stos;
Możemy takie stosy teraz w programie wygenerować:
s:=new stos (100); t:=new stos (911); z:=new stos (5);
przy czym dla każdej generacji obiektu klasy stos wykonywana jest odpowiednia inicjalizacja (a więc generacja tablicy A odpowiednich rozmiarów). Możemy teraz korzystać z tych stosów, np.:
call s.push (7); (* wstaw 7 do stosu s *)
call t.push (1); (* wstaw 1 do stosu t *)
i:=z.pop; (* pobierz kolejny element ze stosu z *)
itd.

```

Wyrażenie postaci *s.push* nazywa się dostępem zdalnym (*remote access*). Jeżeli obiekt klasy *stos* został wygenerowany (*new stos*) i jego adres został przesłany na zmienną tego typu (*s:=new stos*), to z atrybutów tego obiektu możemy korzystać za pomocą dostępu zdalnego.

Zauważmy, że klasa, której jedynymi atrybutami są zmienne, odpowiada PASCALOWEMU rekordowi.

## PREFIKSOWANIE

Klasy i prefiksowanie wprowadzone zostały po raz pierwszy w SIMULI-67 [1]. Niestety, programiści, którzy nie zetknęli się z SIMULA, rzadko wiedzą cokolwiek o tych pojęciach. Ponadto SIMULA, ze względu na implementacyjny, nakłada bardzo silne ograniczenia na stosowanie klas i prefiksowania. Wszystkie te przyczyny złożyły się na to, że te dwa niezwykle silne i użyteczne narzędzia nie zostały wprowadzone do innych języków programowania. Mamy nadzieję, że LOGLAN adaptując pozytywne cechy SIMULI, a ponadto znosząc istotne jej ograniczenia, rozwinię techniki i metodologię programowania.

Co to jest prefiksowanie? Jest to pewna metoda rozszerzania modułów. Najlepiej rozpocząć od prostego przykładu.

```

Mamy w programie deklarację klasy dowód:
unit dowód: class (nazwisko: string, wiek: integer);
end dowód;
napiszmy jej rozszerzenie o nowe parametry, np.:
unit ndowód: dowód class (imię: string, płeć: boolean);
end ndowód;

```

Wówczas dla zmiennych z typu *dowód* i *t* typu *ndowód* odpowiednie generacje będą postacią np.:  
*z:=new dowód („kreczmar”, 37);*  
*t:=new ndowód („kreczmar”, 37, „antoni”, true);*

czyli parametry formalne klasy *ndowód* będą dołączone do parametrów formalnych klasy *dowód*.

Jeżeli chcemy rozbudować dalej klasę *ndowód* o nowe atrybuty, możemy klasą *ndowód* prefiksować inne klasy. Na przykład:

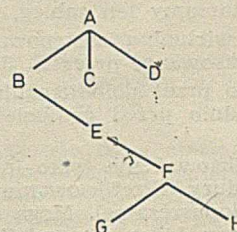
```

unit rdowód: ndowód class;
var liczba_dzieci: integer;
var miejsce_urodzenia: string;
end rdowód;

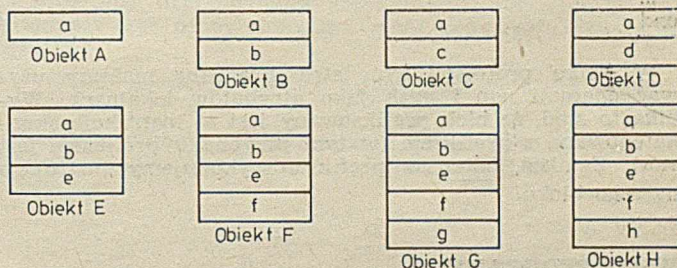
```

Prefiksowanie pozwala w programie utworzyć hierarchię klas w postaci drzewa. Korzeniem tego drzewa jest klasa bez prefiksu, natomiast klasa *A* jest poprzednikiem w tym drzewie klasy *B*, jeżeli *A* prefiksuje *B*.

Rozważmy następującą strukturę prefiksową:



W tym przykładzie klasa *H* ma ciąg prefiksowy *A, B, E, F, H*, czyli wszystkie atrybuty klas *A, B, E, F* będą także atrybutami klasy *H*. Niech *a, b, ..., h* oznaczają przykładowe jedyne zmienne zadeklarowane w tych klasach, odpowiednio w *A, B, ..., H*. Struktury obiektów takich klas można przedstawić w sposób następujący:



Niech *Ra, Rb, ..., Rh* oznaczają teraz zmienne referencyjne, odpowiednio typów *A, B, ..., H*. Zatem następujące sposoby użycia zdalnego dostępu będą poprawne:  
*Ra.a, Rb.b, Rg.g, Rg.f, Rh.h, Rh.f, Rh.e, Rh.b, Rh.a* itp.

Wartość zmiennej *Ra* może jednak nie tylko wskazywać na obiekt klasy *A*, ale również na obiekt dowolnej klasy prefiksowanej przez *A*, np. *Ra:=new G* jest instrukcją poprawną. Wówczas atrybuty *b, e, f, g* tego obiektu będą niedostępne w sposób zdalny, tzn. wyrażenia:

```

Ra.b, Ra.e, Ra.f, Ra.g

```

są niedozwolone, albowiem zmienna referencyjna *Ra* może wskazywać na obiekt typu *A*, który takich atrybutów w ogóle nie posiada. Aby można było jednak użyć tych atrybutów, wprowadzono do języka zmianę kwalifikacji typu:

```

Ra qua G

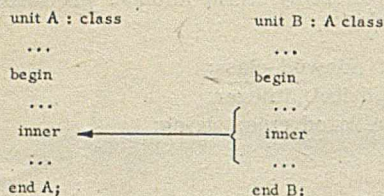
```

i wówczas, jeżeli *Ra* wskazuje na obiekt *G*, to wyrażenie takie jest typu *G* (jeżeli *Ra* nie wskazuje na obiekt klasy *G*, to otrzymujemy błąd w trakcie wykonania programu). Dostęp do atrybutów *b, e, f, g* można więc uzyskać w następujący sposób:  
*Ra qua G.b, Ra qua G.e* itd.

Dotychczas pokazaliśmy, jak wykonuje się konkatencja atrybutów. Ale przecież pamiętamy, że klasy mogą posiadać także instrukcje. Otóż instrukcje są też konkatencowane, a do tego celu służy słowo kluczowe *inner*. Może się ono pojawić tylko raz, w dowolnym miejscu wśród instrukcji klasy. Jeżeli klasa ta prefiksuje jakąś inną kla-



se, to w tej nowej klasie obowiązują wszystkie instrukcje z prefiksu, ze słowem inner zastąpionym przez instrukcje klasy prefiksowanej:



Ciąg instrukcji klasy B definiujemy następująco. Bierzemy wszystkie instrukcje klasy A, zamiast inner w A wstawiamy wszystkie instrukcje klasy B, i to wszystko razem stanowi ciąg instrukcji klasy B (inner w B jest równoważne instrukcji pustej). Jeżeli B prefiksuje jakąś inną klasę, to powtarzamy ten zabieg, przy czym inner w B jest instrukcją wirtualną zastępowaną przez ciąg instrukcji klasy prefiksowanej przez B. Jeżeli inner nie występuje w klasie, to przez domniemanie przyjmuje się, że występuje bezpośrednio przed symbolem end tej klasy.

Klasy mogą prefiksować nie tylko inne klasy, ale także funkcje oraz procedury. (Prefiksowanie współprogramów i procesów też jest dopuszczalne). Mogą one także prefiksować bloki. Rozważmy klasę stos zdefiniowaną uprzednio. Poniżej pokazujemy, jak można tej klasy użyć do prefiksowania bloku:

```

pref stos (1000) block
var ...
begin
call push (50); ...
i := pop;
...
end

```

W bloku prefiksowanym strukturą stos możemy używać operacji pop i push jako atrybutów lokalnych. Wynika to stąd, że blok prefiksowany jest to instrukcja skatelowana z prefiksem (w tym przypadku prefiksem jest stos). Zamiast inner w prefiksie wykonujemy instrukcje naszego bloku.

## WSPÓLPROGRAMY

Współprogram (ang. *coroutine*) można potraktować jako uogólnienie procedury lub klasy. Obiekt współprogramu jest to taki obiekt, z którego sterowanie może wychodzić (bez zniszczenia jego pola danych), a następnie wracać do niego i kontynuować obliczenia od miejsca wyjścia. Rozważmy najpierw zwykłą klasę, w której po instrukcji return znajdują się jakieś inne, nie wykonane jeszcze instrukcje. Klasa taka może być zadeklarowana jako współprogram. Z chwilą wykonania instrukcji powrotu return sterowanie lokalne współprogramu zatrzyma się na tej instrukcji, ale będzie gotowe do wykonania dalszych instrukcji współprogramu. Jeżeli mamy w języku możliwość przekazania sterowania do takiego zawieszzonego współprogramu, to można kontynuować jego obliczenia.

Instrukcja przekazująca w ten sposób sterowanie ma w języku LOGLAN postać następującą:

```
attach (X)
```

gdzie X jest zmienną referencyjną wskazującą na współprogram, do którego sterowanie chcemy przekazać. Ogólnie, współprogram deklarujemy podobnie jak klasę, zastępując słowo class słowem *coroutine*. Jest on generowany jak zwykła klasa, ale od chwili wykonania instrukcji return może być albo aktywny, albo zawieszony. Każde wznowienie działania zawieszzonego współprogramu powoduje kontynuację jego obliczeń od miejsca, gdzie sterowa-

nie lokalne zostało zawieszono. Każde przekazanie sterowania do zawieszzonego współprogramu powoduje jednocześnie zawieszenie aktywnego współprogramu. Program główny jest także współprogramem o nazwie main. Jeżeli zostanie zawieszony, to można do niego przekazać sterowanie instrukcją attach (main).

W języku LOGLAN występuje także druga operacja przekazująca sterowanie pomiędzy współprogramami, mianowicie detach. Potrzeba wprowadzenia takiej operacji wynika z następującego spostrzeżenia. Bardzo często po przekazaniu sterowania do współprogramu X za pomocą attach (X) i po wykonaniu pewnych czynności przez ten współprogram, chcemy zwrócić sterowanie tam, skąd ono przyszło. Co więcej — nie znamy częstokroć adresata, tzn. współprogramu, który wykonał to attach (X). Do tego celu służy właśnie instrukcja detach. Ma ona zatem podobny sens, co return w procedurach i funkcjach. Niemniej trzeba pamiętać, że jest to zupełnie inny, mniej bezpieczny mechanizm. Tutaj programista może dowolnie „zapęlić” ślady powrotu po współprogramach, np. przyłączając nawzajem dwa współprogramy:

```
w X attach (Y)
```

```
w Y attach (X)
```

i wówczas wykonanie detach w X zwróci sterowanie do Y, a wykonanie detach w Y zwróci sterowanie do X.

Współprogramy odgrywają ważną rolę w zadaniach symulacyjnych. Klasa *Simulation*, zaprogramowana do tych celów po raz pierwszy w SIMULI-67, korzysta ze współprogramów w sposób niezwykle istotny. LOGLAN oczywiście także umożliwia symulację. LOGLANOWA klasa *Simulation* jest zrealizowana w sposób nowoczesny i pozwala efektywnie rozwiązywać różnorodne zadania symulacyjne.

## TYPY FORMALNE

Typy formalne pozwalają parametryzować moduły ze względu na dowolny złożony typ danych. Rozważmy przykład:

```

unit Gsort: procedure (type T; A: arrayof T; function
less (x, y : T): boolean);
(* sortuje tablicę A ze względu na funkcję less *)
...
end Gsort;

```

W powyższym nagłówku procedury mamy parametr T — typ formalny. Parametrem aktualnym może być dowolny typ złożony. Zauważmy także, że A jest tablicą elementów typu formalnego T, oraz że funkcja boolowska „less” określająca porządek w zbiorze elementów tego typu też ma parametry formalne x, y typu formalnego T.

Rozważmy deklarację typu dowód oraz tablicę A elementów tego typu. Możemy tę tablicę uporządkować ze względu na atrybut nazwisko definiując odpowiednią procedurę porządkującą:

```

unit mniejsze: function (t, u: dowód): boolean;
begin
result: = t. nazwisko <= u. nazwisko
end

```

a następnie wywołując procedurę Gsort: call Gsort (dowód, A, mniejsze).

Jeżeli chcemy te tablice uporządkować ze względu na wiek, należy zadeklarować inną funkcję porządkującą: unit wcześniej: function (t, u: dowód): boolean;

```
begin
```

```
result: = t. wiek <= u. wiek
```

```
end;
```

a następnie wywołać: call Gsort (dowód, A, wcześniej).

## LITERATURA

[1] Dahl O. J., Myhrhaug B., Nygaard K.: Common Base Language. NCC s-22, October 1970.



# Dwie metody badania sprawności systemów komputerowych

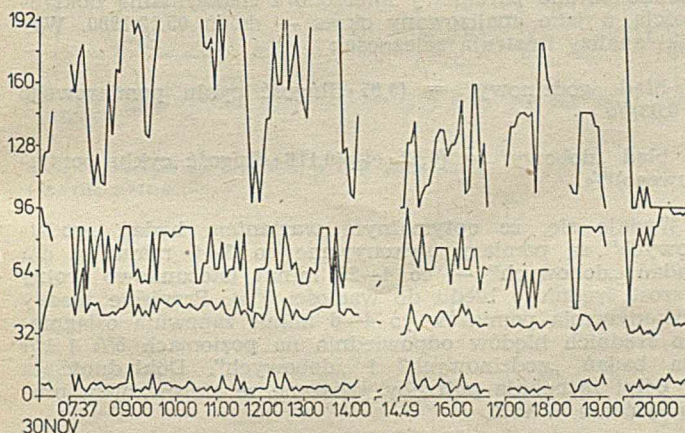
Artykuł przedstawia dwie praktyczne metody badania sprawności systemów komputerowych. Prezentuje on wyniki badań sprawności Wielodostępnego Abonenckiego Systemu Komputerowego (WASK), jakie w 1980 r. przeprowadzono za pomocą tych metod w poznańskiej Akademii Ekonomicznej. Obie metody przedstawione były w odrębnych referatach na drugiej międzynarodowej konferencji naukowo-technicznej pn. „Niezawodność i eksploatacja systemów komputerowych — RELCOMEX'81” (Książ, wrzesień 1981).

Prezentowane wyniki dotyczą systemu WASK w następującej konfiguracji:

- procesor ODRA 1305 z pamięcią operacyjną o pojemności 96 K słów
- pamięci zewnętrzne: bębnowa — PBS 304, dyskowa — EC 5052 (6 szt.) i taśmowa — PT 3M (6 szt.)
- sieć terminali typu DZM 180 KSRE (6 szt.) oraz monitory ekranowe VT 340 (12 szt.)
- system operacyjny GEORGE 3 (wersja 8.64).

## METODA MONITORÓW PROGRAMOWYCH

Przy analizie sprawności systemu WASK metodą monitorów programowych wykorzystano standardową możliwość systemu operacyjnego GEORGE 3 gromadzenia informacji statystycznych o pracy systemu w regularnych odcinkach czasowych (tzw. system PERFORMANCE [3]). W Akademii Ekonomicznej w Poznaniu opracowano specjalny program wydruku oraz analizy zbioru informacji statystycznych gromadzonych przez system. Program ten



Rys. 1. Wykorzystanie pamięci operacyjnej systemu WASK  
1.FREE 2.EXEC+G'3 3.CHAP 4.SUMA

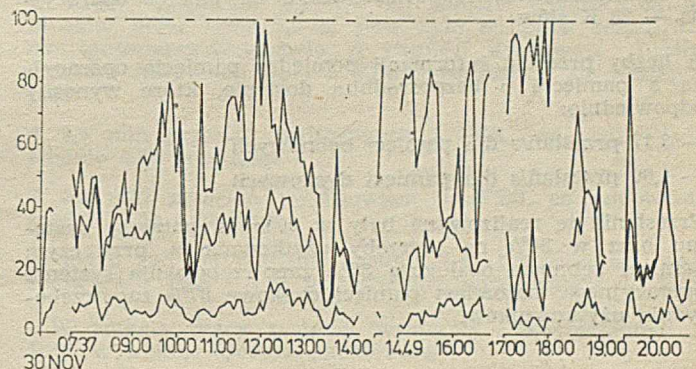
1) Akademia Ekonomiczna, Uczelniany Ośrodek Przetwarzania Informacji, ul. Marchlewskiego 146/150, 60-967 Poznań

posiada jednocześnie możliwość wyprowadzenia wielu zestawień graficznych na autokreślarkę EC 7054, przyłączonej bezpośrednio do systemu WASK.

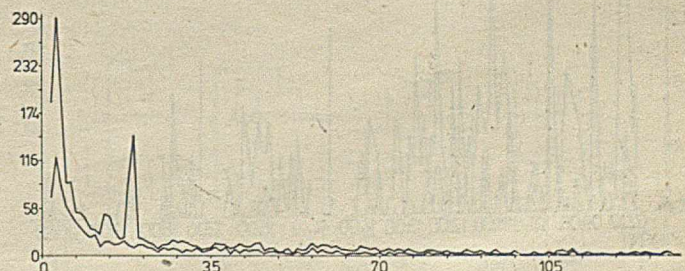
Zbrane i przetworzone dane statystyczne o pracy systemu komputerowego stanowią materiał obrazujący rzeczywiste wykorzystanie takich czynników, jak: czas pracy procesora, zajętość pamięci operacyjnej lub przepustowość kanałów przesyłania informacji pomiędzy pamięcią operacyjną a pamięciami bębnowymi i dyskowymi.

Pamięć operacyjna systemu WASK wykorzystywana jest przez:

- alokator wolnego obszaru pamięci operacyjnej
- część rezydującą systemu operacyjnego GEORGE 3
- część wymienną systemu operacyjnego GEORGE 3
- programy użytkowników (biblioteczne lub własne), natomiast czas pracy procesora może być wykorzystywany przez:
- system operacyjny — na wewnętrzne potrzeby organizacyjne
- użytkowników — dla realizacji programów (w tym — praca modułów wymiennych systemu GEORGE 3).

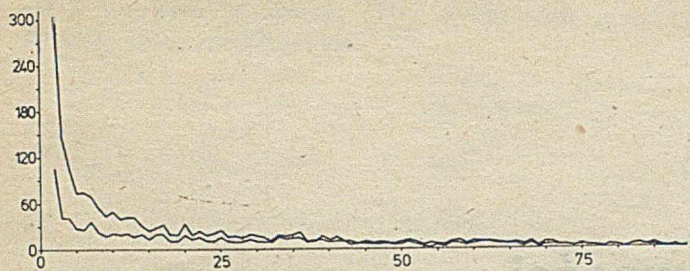


Rys. 2. Wykorzystanie mocy procesora systemu WASK  
1.EXEC 2.G'3 3.PROG



Rys. 3. Wykorzystanie procesora przez zadania





Rys. 4. Czas trwania zadań

Sredniomiesięczne wykorzystanie tych zasobów można scharakteryzować za pomocą następujących parametrów:

- mocy obliczeniowej procesora — wykorzystana była przez system w 7% i przez użytkowników w 33% (w tym przez programy własne w 15%)



Rys. 5. Wykorzystanie pamięci operacyjnej przez zadania

- wielkości obszarów pamięci operacyjnej, które wynosiły odpowiednio:

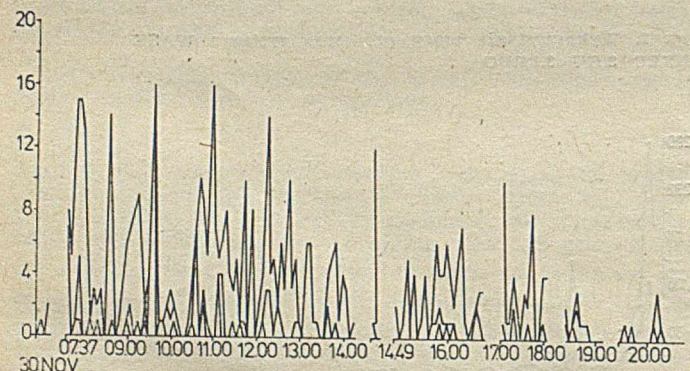
- części wolnej (alokatora pamięci) — 7 K słów
- części stałej systemu operacyjnego — 34 K słów
- części wymiennej systemu operacyjnego — 38 K słów
- programów użytkowych — 17 K słów

- niezaspokojonego zapotrzebowania na pamięć operacyjną — 23 K słów

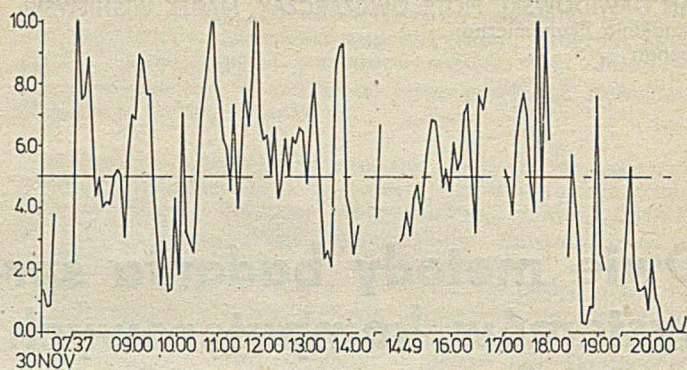
- liczby przesłań informacji pomiędzy pamięcią operacyjną a pamięcią o bezpośrednim dostępie, które wynosiły odpowiednio:

- 3,37 przesłań/s dla pamięci bębnowych
- 4,66 przesłań/s dla pamięci dyskowych

Przesłania te realizowane były w 50% na potrzeby systemu oraz w 50% na potrzeby użytkowników, przy czym pamięci bębnowe obsługują 85% zapotrzebowania systemu operacyjnego, natomiast pamięci dyskowe 87% zapotrzebowania użytkowników.



Rys. 6. Kolejki transmisji do kanałów  
1. 60 2. 50 3. 30



Rys. 7. Współczynnik transmisji rozdzielców G³

Bardziej ogólnie zmienność mierzonych parametrów można przedstawić za pomocą następujących zależności:

- system realizuje średnio 3,97 zadania zdalnego i 2,63 zadania wsadowego jednocześnie
- jedno zadanie użytkownika angażuje średnio 5% mocy obliczeniowej procesora
- jedno zadanie użytkownika angażuje średnio 5800 słów pamięci operacyjnej i powoduje powiększenie obszaru części stałej systemu operacyjnego o ok. 800 słów.

Potrzeby wewnętrzne systemu operacyjnego można natomiast scharakteryzować za pomocą następujących zależności:

- procent angażowanej przez system operacyjny mocy obliczeniowej procesora wzrasta wykładniczo wraz ze wzrostem liczby przesłań informacji na sekundę pomiędzy pamięcią operacyjną a pamięcią o bezpośrednim dostępie
- wielkość obszaru na część wymienną systemu operacyjnego jest „wymienna” z liczbą przesłań na sekundę pomiędzy pamięcią operacyjną a pamięcią o bezpośrednim dostępie (97% przesłań na potrzeby systemu).

Dla zbadania wpływu częstości wykonywania pomiarów na dokładność uzyskiwania wyników wykonano eksperyment polegający na utworzeniu zbioru pomiarów z maksymalną możliwą częstością (pomiar co minutę) oraz programowej symulacji wykonywania pomiarów w innych odcinkach czasowych. Analizy tej dokonano osobno dla charakterystyk godzinowych i dobowych. Jako miarę dokładności poszczególnych pomiarów obrano średni procentowy błąd względny pomiaru 16 parametrów istotnych (ze względu na prowadzone badania), jako wartości dokładne obrano parametry mierzone z maksymalną dokładnością, a jako analizowany okres — dzień 05.12.1980. Wyniki analizy ilustrują zależności:

- błąd „godzinowy” =  $(0,65 \cdot \text{długość cyklu pomiarowego} + 0,98)\%$

- błąd „dobowy” =  $(0,52 \cdot \exp(0,116 \cdot \text{długość cyklu pomiarowego}))\%$

Wydaje się, że optymalnym wariantem badań „godzinowych” są pomiary wykonywane co 3–6 minut, a dla badań „dobowych” — co 4–20 minut (stosunkowo wolny wzrost średniego błędu do wartości 5%). Przyjęcie zasady wykonywania pomiarów co 4–6 minut zapewnia osiągnięcie średnich błędów odpowiednio na poziomach 5% i 1% dla badań „godzinowych” i „dobowych”. Dokładność ta w pełni zaspokaja potrzeby związane ze sterowaniem pracą systemu WASK.

Przeprowadzone badania potwierdziły przydatność metody monitorów programowych do badania sprawności systemu komputerowego. Informacje statystyczne zbierane cyklicznie przez system operacyjny pozwalają precyzyjnie określić stan wykorzystywania takich czynników, jak: czas pracy procesora, zajętość pamięci operacyjnej lub przepustowość kanałów przesyłania informacji pomiędzy pamięcią operacyjną a pamięciami o bezpośrednim dostę-



pie. Ogólne zależności uzyskane z informacji o wykorzystaniu tych czynników pozwalają łatwiej dobrać wartości poszczególnych parametrów instalacyjnych systemu GEORGE 3 i — co jest z tym nieodłącznie związane — ułatwiają identyfikację przeciążeń systemu. Praktyczna zdolność sterowania wydajnością systemu WASK ograniczona jest jednak w bardzo dużym stopniu znacznym przeciążeniem pamięci operacyjnej, wynoszącym w skali miesiąca średnio 124% (!).

### METODA ANALIZY DZIENNIKA

Prezentowane wyniki analizy sprawności systemu WASK metodą analizy-dziennika opierają się na informacjach zebranych przez:

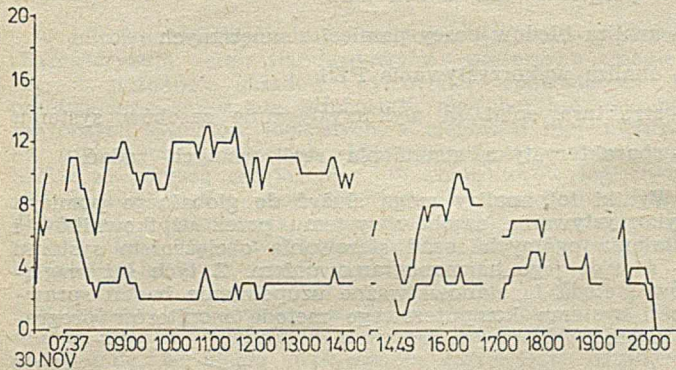
- system operacyjny GEORGE 3 (w dzienniku pracy systemu i zbiorach UNJAMMER)
- system rozliczeń użytkowników systemu operacyjnego GEORGE 3 (opracowany i rozpowszechniany przez ZETO Wrocław).

Po upływie analizowanego okresu zebrane informacje zostały przetworzone przez:

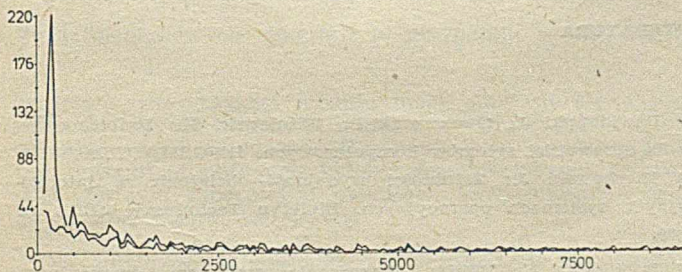
- standardowy system analizy pracy urządzeń RSJ8 (wersja 110)
- standardowy program wydruku zbioru UNJAMMER
- opracowany w Akademii Ekonomicznej w Poznaniu zestaw programów analizy rozliczeń systemu komputerowego, przystosowany do wyprowadzenia wyników na auto-kreślarkę EC 7054.

Wyniki badania sprawności systemu zaprezentowano w następujących grupach zagadnień:

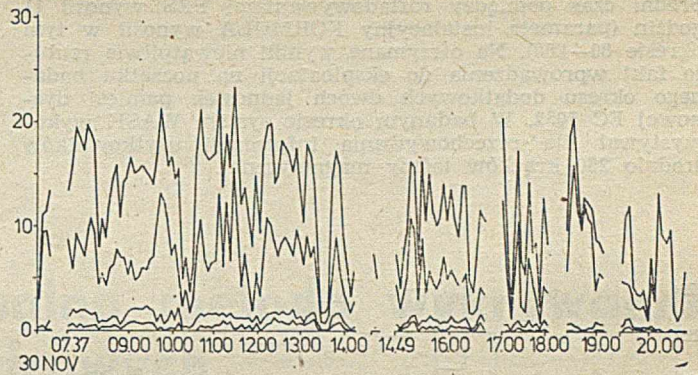
- analiza pracy urządzeń pamięci zewnętrznych
- analizy pracy i wykorzystania Pamięci Zbiorów Systemu
- analiza charakterystyk pracy systemu.



Rys. 8. Obciążenie systemu WASK zadaniami  
1.BACK 2.MOP



Rys. 9. Liczba transmisji z/do PZS w zadaniach



Rys. 10. Wykorzystanie pamięci o bezpośrednim dostępie systemu WASK  
1.WYMIANA 2.CHAP Z ED 3.PZS+ON LINE 4.CHAP Z DR

Analiza pracy urządzeń pamięci zewnętrznych systemu WASK wykazała, że w badanym okresie system zrealizował:

- 5,2 mln przesłań z pamięciami bębnowymi, w trakcie których wystąpiło 9 przesłań błędnych
- 3,8 mln przesłań z pamięciami dyskowymi, w trakcie których wystąpiło ok. 3,9 tys. przesłań błędnych
- 4,5 mln przesłań z pamięciami taśmowymi, w trakcie których wystąpiło 700 przesłań błędnych.

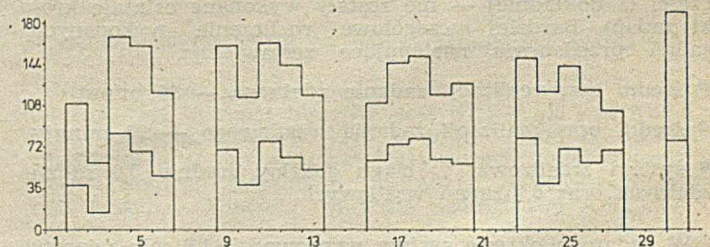
W trakcie pracy wykonano ponad 2,3 tys. zmian krążków taśmy magnetycznej w jednostkach pamięci taśmowej oraz ok. 60 zmian pakietów dyskowych w jednostkach pamięci dyskowej. Rozkład ilościowy przesłań dla jednostek pamięci taśmowej i dyskowej był równomierny ( $\pm 10\%$ ), natomiast rozbieżności w wielkości stopy błędów dla poszczególnych jednostek wynosiły dla pamięci taśmowej — 1:2,5, natomiast dla pamięci dyskowej — 1:60, osiągając wartość 4400 błędów na mln zrealizowanych przesłań.

Pamięć Zbiorów Systemu (PZS) jest podstawowym miejscem przechowywania informacji użytkowników systemu. W badanym okresie wielkość tej pamięci wynosiła ok. 55 M słów, gdzie tylko 5,9 M słów było dostępne bezpośrednio (przechowywane było w pamięci dyskowej). Liczby przesłań informacji do i z PZS kształtowały się następująco:

- 4,3 mln przesłań wejściowych z PZS, co odpowiada średnio 4,68 przesłań/s
- 3,1 mln przesłań wyjściowych do PZS, co odpowiada średnio 3,33 przesłań/s
- jedno zadanie realizuje średnio 1,43 tys. przesłań wejściowych z PZS i 1,04 tys. przesłań wyjściowych do PZS.

W trakcie pracy systemu wystąpiły 23 rozładowania PZS, w trakcie których każdorazowo:

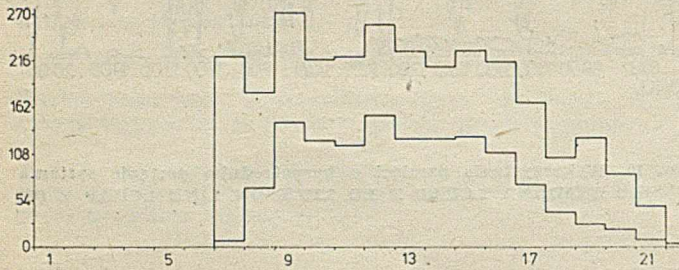
- 42% dostępnej bezpośrednio pojemności PZS przenoszona było do pamięci taśmowej
- 181 dostępnych bezpośrednio zbiorów informacji przenoszonych było do pamięci taśmowej.



Rys. 11. Miesięczny rozkład liczby zadań w poszczególnych dniach



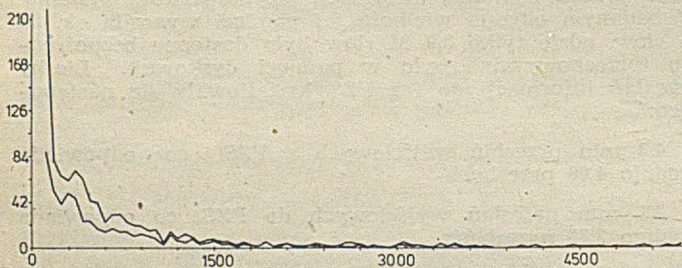
Średni czas pomiędzy rozładowywaniami PZS wynosił 11 godzin (parametr instalacyjny FORMULA wynosił w tym okresie 80—200). Na otrzymane wyniki niewątpliwie rzutowa fakt wprowadzenia do eksploatacji na początku badanego okresu dodatkowych dwóch jednostek pamięci dyskowej EC 5052. W badanym okresie system WASK wykorzystywał dla przechowywania informacji użytkowników średnio 260 krążków taśmy magnetycznej.



Rys. 12. Dzienny rozkład liczby zadań w poszczególnych godzinach

**Analiza charakterystyk pracy systemu.** Rozpocznijmy ją od przedstawienia ogólnej struktury opłat za:

- wykorzystywanie czasu procesora — 16,75%
- wykorzystywanie pamięci operacyjnej — 12,35%
- wykorzystywanie PZS — 36,09%
- drukowanie wyników — 4,44%
- przesyłanie informacji do i z PZS — 16,23%
- korzystanie z sieci terminali — 12,97%
- wykorzystywanie innych zasobów systemu — 1,17%.



Rys. 13. Liczba wydrukowanych stron przez zadania

W badanym okresie system zrealizował 3 tys. zadań (w tym 1,3 tys. zadań realizowanych za pomocą sieci terminali) na potrzeby 113 użytkowników (a dokładniej — na potrzeby 81 użytkowników aktywnych) w ciągu 255,3 użytecznych godzin pracy. Wykonywanie ok. 350 zadań zostało przerwane z powodu awarii całego systemu lub jego części (a dokładniej — nie zostały wykonane ostatnie kroki zadań). Bardziej szczegółowe rozliczenie wykonanych zadań przedstawia następujące zestawienie:

- średni czas realizacji zadania zdalnego — 40 minut
- średni czas realizacji zadania wsadowego — 23 minuty
- system zrealizował w ciągu godziny średnio 5,0 zadań zdalnych oraz 6,7 zadań wsadowych.

W badanym okresie system wyprowadził 1,6 mln wierszy wydruków, co odpowiada średnio 6,3 tys. wierszy w ciągu godziny lub 535 wierszom wydruku na zrealizowane za-

danie. Biorąc średnie wypełnienie strony równe 60 wierszy (pełna strona — 72 wiersze) otrzymujemy miesięczną porcję wydruków równą ok. 27 tys. stron.

W celu oceny dokładności przeprowadzonych badań dokonano porównania uzyskanych wyników z wynikami badań za pomocą metody monitorów programowych. Różnice obrazuje następujące zestawienie:

Parametry	Metody	
	analizy dziennika	monitorów programowych
Użytkowe wykorzystanie czasu procesora	21%	33%
Liczba jednocześnie zrealizowanych zadań zdalnych	3,84	3,97
Liczba jednocześnie realizowanych zadań wsadowych	2,55	2,63
Sumaryczne wykorzystanie pamięci operacyjnej	31 K słów	40 K słów
Sumaryczna liczba przesłań bębnowych	5,2 mln	3,1 mln
Sumaryczna liczba przesłań dyskowych	3,8 mln	4,3 mln

Większość porównywalnych parametrów wskazuje na znaczny (20%) błąd systematyczny, wynikający ze sposobu rozliczania pracy systemu (nie uwzględnia się prac nie zakończonych z powodu awarii sprzętu), natomiast liczba przesłań bębnowych naliczana jest przez obie metody według różnych zasad.

Metoda badania sprawności systemu komputerowego oparta o analizę dziennika pracy systemu, aczkolwiek obciążona znacznym błędem w stosunku do rzeczywistego obciążenia systemu, ma wymierne walory praktyczne z uwagi na:

- łatwość stosowania
- wyrażenie sprawności systemu komputerowego w kategoriach wykonanych prac
- uwzględnianie zagadnień nie objętych metodą monitorów programowych, takich jak:
  - analiza błędów pracy pamięci zewnętrznych
  - analiza wykorzystywania PZS
  - struktura opłat za wykorzystywanie zasobów systemu
  - charakterystyka strumienia realizowanych zadań.

Wyniki tej analizy mogą służyć do globalnego pomiaru wykorzystywania zasobów systemu przez użytkowników i ustalania ogólnych zasad sterowania obciążeniem systemu — a więc jego harmonogramowaniem. Z tych też względów metoda ta stanowi ważne uzupełnienie badań sprawności systemu komputerowego metodą monitorów programowych.

#### LITERATURA

- [1] Dawidowski J., Owczarczak P., Wiśniewski M.: Problemy badania sprawności systemów komputerowych. *INFORMATYKA* 4/1981
- [2] Kierkowski J.: Zależności wydajność—obciążenie w interakcyjnych systemach operacyjnych. *Biuletyn Techniczny MERA* 11/1980
- [3] Publikacja Nr 1300204: Zarządzenie systemem operacyjnym GEORGE 3. Centrum MERA ELWRO, wyd. 2, Wrocław 1977.



# Analiza sygnatur - uniwersalna metoda wykrywania uszkodzeń w układach cyfrowych

Obecny etap komputeryzacji na świecie charakteryzuje się lawinowym wzrostem liczby mini- i mikrokomputerów. Uniwersalność i niskie koszty produkcji seryjnej układów scalonych LSI, a w tym mikroprocesorów — pozwalają na ich masowe zastosowanie w codziennym życiu. Podobnie jak gwałtowny wzrost liczby odbiorników radiowych czy telewizyjnych — tak i gwałtowny wzrost liczby urządzeń cyfrowych wymaga stworzenia odpowiedniego zaplecza serwisowego. Niniejszy artykuł przedstawia uniwersalną metodę wykrywania uszkodzeń tych urządzeń.

W przypadku sprzętu radiotechnicznego metody lokalizacji uszkodzeń są stosunkowo proste. Dysponując przyrządami pomiarowymi oraz schematem urządzenia z naniesionymi nominalnymi wartościami napięć, prądów i oscylogramami w charakterystycznych punktach, można sprawdzić czy dane układy elektroniczne pracują właściwie. Poprawność działania poszczególnych bloków urządzenia sprawdzamy posuwając się od wyjścia w kierunku wejścia. Jeżeli na wyjściu danego podzespołu (lub elementu) zmierzone wartości nie odpowiadają nominalnym, a na wejściu układu są one zgodne z wartościami odczytanymi ze schematu, to wymianie podlega dany podzespół (element).

W odróżnieniu od techniki analogowej, układy cyfrowe charakteryzują się standardowymi poziomami napięć. Wzrost złożoności układów cyfrowych spowodował, że przetwarzana przez nie informacja ma charakter nie pojedynczych sygnałów logicznych, a ciągów o niejednokrotnie znacznych długościach. Identyfikacja tego typu sygnałów za pomocą oscyloskopu lub analizatorów logicznych jest sprawą bardzo skomplikowaną, a często niemożliwą. Znalazienie uszkodzenia wymaga dokładnej znajomości pracy układu i nawet doświadczonemu inżynierowi zadanie to może zająć dużo czasu.

Najczęściej stosowana metoda naprawy polega na wymianie modułu — płytki drukowanej, co do której istnieje podejrzenie, że uległa uszkodzeniu. Lokalizacja uszkodzonego modułu, a nie elementu — pozwala na znaczne skrócenie czasu naprawy. Tego rodzaju metoda pociąga za sobą dodatkowe koszty związane z:

- koniecznością zakupu kompletu zapasowych modułów
- przechowywaniem i katalogowaniem modułów
- permanentnym uaktualnieniem biblioteki modułów
- zakupem nowych modułów w przypadku zmiany sprzętu.

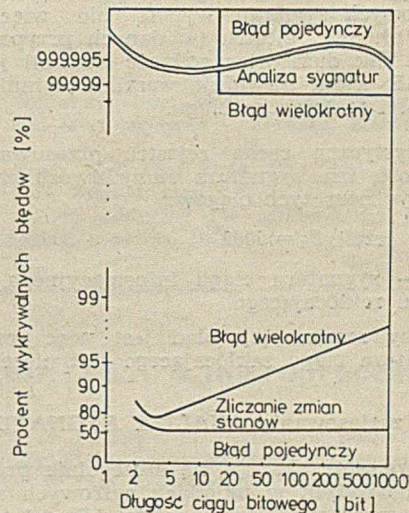
Naprawa uszkodzonego modułu może odbywać się u producenta lub we własnym zakresie — przez personel obsługujący dane urządzenie. Ponieważ okres naprawy może być dość długi pożądane byłoby przechowywanie podwójnego zestawu zapasowych modułów, co oczywiście pociąga za sobą dodatkowe koszty.

Dla zmniejszenia nakładów na eksploatację nie wystarczy dokładne wstępne testowanie elementów, trzeba również stworzyć nową strategię obsługi układów cyfrowych.

Informacja dostarczana przez obecnie stosowane przyrządy kontrolne charakteryzuje się zwykle nadmiarowością albo też jest zbyt skąpa, aby można ocenić prawidłowość funkcjonowania układu.

W celu przetestowania układu cyfrowego należy podać na jego wejścia określone kombinacje sygnałów logicznych. Istotne jest, aby powodowały one możliwie częstą zmianę sygnałów na wyjściu. Liczba kombinacji testowych wzrasta wg funkcji wykładniczej ze wzrostem stopnia złożoności urządzenia. Dla określenia poprawności pracy układu należy stwierdzić, czy jego reakcje na testy (ciągi zerowejedynkowe) odpowiadają wzorcowym.

Naturalnym dążeniem było skondensowanie informacji dotyczącej prawidłowości funkcjonowania układu, a zawartej w ciągu bitowym. Najczęściej obecnie stosowaną metodą jest „zliczanie zmian stanów”. Pozwala ona zastąpić ciąg „0” i „1” liczbą zmian stanów logicznych. Jej główną wadą jest mała wiarygodność (rys. 1) [4].



Rys. 1. Porównanie efektywności wykrywania błędów metodą zliczania zmian stanów oraz metod analizy sygnatur (długość rejestru — 16 bitów)

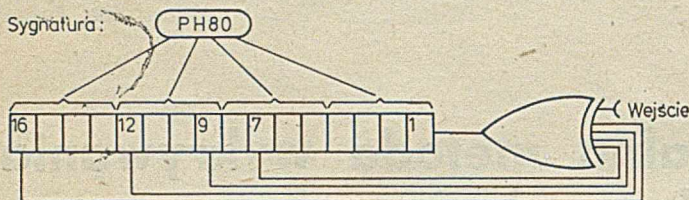
## ANALIZA SYGNATUR — nowa metoda testowania

Nową metodą pozwalającą na skondensowanie informacji zawartej w ciągu bitowym jest analiza sygnatur (AS), zaproponowana w 1977 r. przez firmę HEWLETT-PACKARD. U jej podstawy leży powszechnie stosowany sposób kontroli informacji zapisanych na dyskach magnetycznych — kontrola cykliczna nadmiarowa (CRC).

Zasadniczą częścią przyrządu, w którym wykorzystano tę koncepcję, jest rejestr przesuwający o długości 16 bitów, ze sprzężeniami zwrotnymi (w danym przypadku są



to pozycje 7, 9, 12, 16). Ich zawartość jest sumowana — zgodnie z zasadą modulo 2 — z przychodzącymi na wejście rejestru bitami „skracanego ciągu”. Dzięki sprzężeniom zwrotnym, końcowa zawartość rejestru jest funkcją ciągu



Rys. 2. Szesnastobitowy, pseudolosowy rejestr przesuwający, wyświetlający swoją zawartość w postaci sygnatury

wejściowego, który był przyjęty w ściśle określonym czasie — w tzw. oknie (rys. 2). Tego typu rejestr, pracujący właściwie jako generator liczb losowych, może znajdować się w  $2^{16}-1 = 65535$  różnych stanach. Końcowa zawartość rejestru przekodowana w cztery szesnastkowe cyfry, wg następującego kodu 0, 1, 2, ..., 9, A, C, F, H, P, U tworzy sygnaturę.

Ograniczenie długości rejestru do czterech cyfr jest kompromisem pomiędzy dokładnością i prostotą użytkowania. Produkowany seryjnie przez firmę HEWLETT-PACKARD analizator sygnatur 5004A wyświetla zawartość 16-bitowego rejestru w postaci zmodyfikowanych czterech cyfr szesnastkowych.

Prawdopodobieństwo wykrycia błędu wielokrotnego w ciągu o długości przekraczającej 16 bitów wynosi  $99,998\%$ , w pozostałych ciągach —  $100\%$ . Wykrywalne są również wszystkie pojedyncze błędy (niezależnie od długości ciągu). Wiarygodność analizy sygnatur jest daleko większa niż dotychczas stosowanych metod (rys. 1). Dla dwóch różnych ciągów binarnych prawdopodobieństwo otrzymania jednakowych sygnatur wynosi nie więcej niż  $2^{-n}$  gdzie  $n$  — długość rejestru (w danych przypadku  $2^{-16} = 0,000015$ ). Tak duża wiarygodność kontroli jest szczególnie cechą AS, która likwiduje wszelkie ograniczenia dotyczące długości testującego ciągu.

Charakterystyczną cechą rejestru przesuwającego jest jego liniowość, tzn. sygnatura sumy dwóch ciągów równa się sumie sygnatur tych ciągów:

$$S_z + S_1 = S_2; \text{ jeżeli } S_z = 0000$$

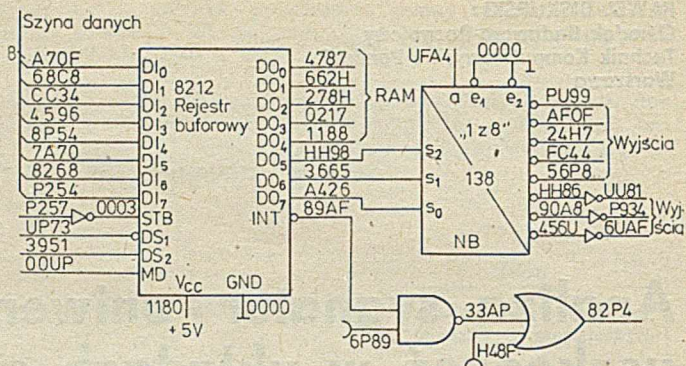
gdzie:  $S_1$  — sygnatura ciągu informacyjnego,  $S_2$  — sygnatura ciągu zakłócającego.

Ciąg binarny zawierający błąd jest więc rozpoznawalny, jeżeli sygnatura ciągu zakłócającego jest niezerowa.

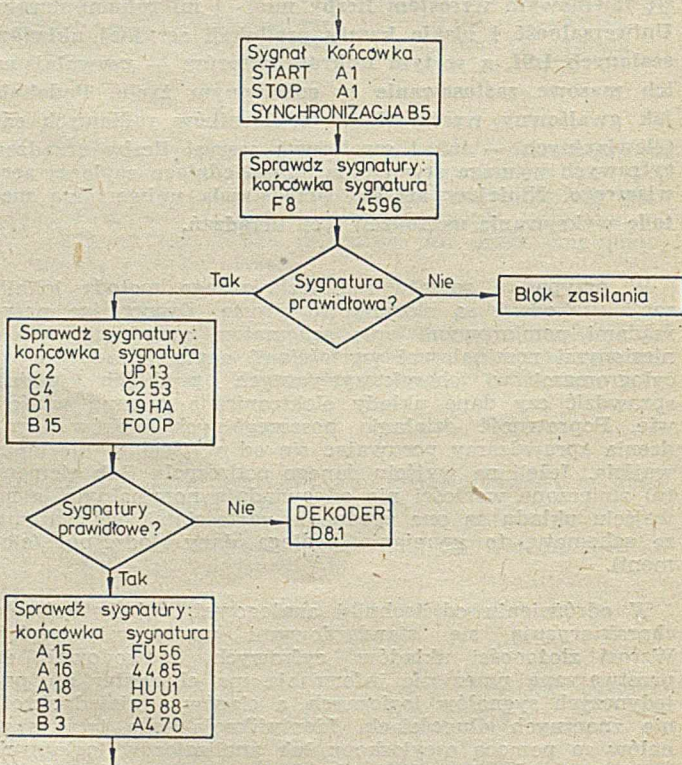
### Praktyczne zastosowanie ANALIZY SYGNATUR

Analiza sygnatur okazuje się szczególnie przydatna przy lokalizacji uszkodzeń w układach cyfrowych zbudowanych na mikroprocesorach. Główną zaletą AS jest to, że sprawdzanie urządzenia odbywa się z normalną, eksploatacyjną szybkością w czasie jego pracy. Na wejście układu podajemy kombinacje testowe. Ich źródłem mogą być pamięci PROM lub — w przypadku układów mniej skomplikowanych — liczniki dwójkowe, pozwalające generować wszystkie możliwe kombinacje wejściowe.

Jednoznaczne przyporządkowanie sygnatury każdemu ciągowi pozwala na opracowanie schematów, na których zamiast napięć, prądów, oscylogramów są naniesione sygnatury (rys. 3). W celu jeszcze większego ułatwienia pracy personelowi obsługi można opracować sieć działań, która szczegółowo — w zależności od znaczeń sygnatur, w charakterystycznych punktach — informuje o uszkodzonym elemencie. Kontrolę urządzenia można również zacząć od punktu będącego jego umownym środkiem i sprawdzać dalej poprawność sygnatur — korzystając z metody dzielenia dwójkowego, tzn. dzieląc wadliwie działającą strukturę na coraz mniejsze bloki (rys. 4).



Rys. 3. Schemat ideowy układu cyfrowego z naniesionymi sygnaturami



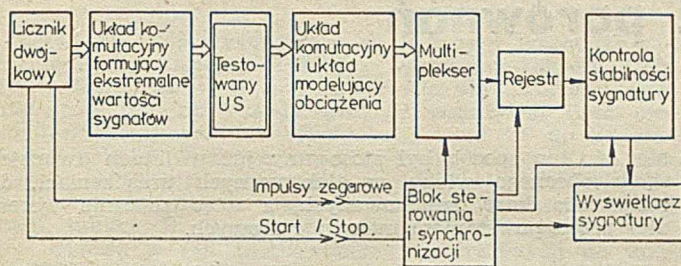
Rys. 4. Sieć działań używana w celu lokalizacji uszkodzenia w układzie cyfrowym — fragment instrukcji dla personelu serwisowego

W przypadku układów mikroprocesorowych — jeżeli uszkodzenie leży w układach zewnętrznych systemu (układy wejścia-wyjścia, klawiatura, itp.), to testy można generować z pamięci ROM, sterowanej szyną adresową samego mikroprocesora. Jeżeli nieprawidłowo pracuje jądro systemu, to AS pozwala również na lokalizację uszkodzenia. W tym celu na wejście mikroprocesora należy podać instrukcję NOP (no operation). Mikroprocesor spełnia wtedy właściwie funkcję licznika, którego wyjściami są szyny adresowe. Rejestrując na nich sygnatury, testujemy tym samym podstawowe układy mikroprocesora. W następnej kolejności sprawdzamy poprawność pracy pamięci (w której zapisany jest program stymulujący testy), prawidłowość wykonywania rozkazów, itd. W ten sposób układ mikroprocesorowy może testować sam siebie.

Ułatwienie czynności serwisowych pociąga za sobą oczywiście dodatkowe koszty. Inżynier-konstruktor powinien uwzględnić na etapie projektowania takie elementy, jak: dodatkową pamięć ROM z testem (jeżeli w podstawowej pamięci systemu nie ma miejsca na jego zapisanie), dodatkowe łączniki, przełączniki, gniazda wyprowadzeniowe



(które pozwalałyby na łatwe odizolowanie badanych układów), rozzerwanie sprzężeń zwrotnych, itp. Dodatkowe koszty zostaną jednak w pełni zbilansowane dzięki zmniejszonym wydatkom na naprawę i obsługę, a także na szkolenie personelu serwisowego.



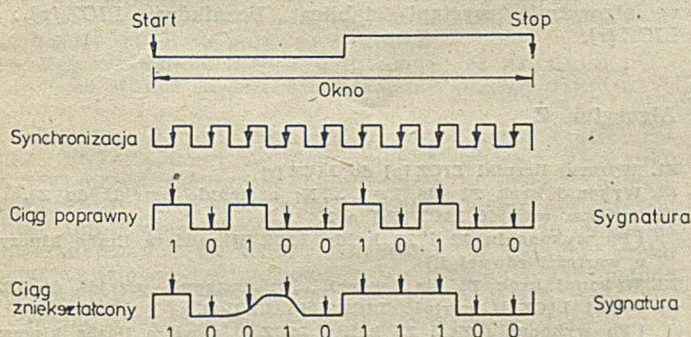
Rys. 5. Schemat blokowy przyrządu do parametryczno-funkcjonalnej kontroli scalonych układów cyfrowych

Analiza sygnatur może być również wykorzystywana do lokalizacji uszkodzeń w urządzeniach zaprojektowanych bez uwzględnienia powyższych wymagań. W tym celu oprócz analizatora sygnatur należy dysponować dodatkowym zespołem generatorów kombinacji testowych, tzn: licznikami dwójkowymi, pamięciami PROM, a także kompletem łączników, pozwalających na przyłączenie generatorów do wejść testowanych modułów.

Ze względu na niski koszt i uniwersalność, przyrząd będący połączeniem analizatora sygnatur oraz generatora testów doskonale nadaje się do funkcjonalno-parametrycznej kontroli cyfrowych układów scalonych dowolnej skali integracji — w warunkach polowych, a więc poza warsztatem naprawczym. Schemat blokowy przyrządu przedstawia rysunek 5. Oba bloki, z których składa się przyrząd, mogą być wykorzystywane niezależnie. Rolę generatora testów pełni licznik dwójkowy. Jego wyjścia poprzez płytkę komutacyjną, są połączone z wejściami badanego układu scalonego (US). Użycie licznika dwójkowego, który generuje wszystkie możliwe słowa binarne, daje pewność, że wśród nich jest konieczny i wystarczający zbiór kombinacji testowych. Dla większości US liczba wejść wynosi 10—12. Jeśli założyc, że licznik jest taktowany sygnałem 5 MHz, to czas potrzebny na generację pełnego zbioru testów równa się

$$\frac{2^{12}}{5 \cdot 10^{-6}} = 10^{-3} s$$

W charakterze rejestratora reakcji wyjść badanego US został użyty analizator sygnatur. Z punktu widzenia efektywności testowania, zastąpienie minimalnego zbioru testów zbiorem pełnym praktycznie nie ma żadnego znaczenia — manipulacje operatora związane z przyłączeniem testowanego układu trwają znacznie dłużej. Jedynym warunkiem poprawności testowania jest stabilność okna



Rys. 6. Zasada wykrywania przekłamań w ciągach bitowych za pomocą analizy sygnatur

(tzn. czas trwania jednego cyklu testowania), a także powtarzalność testów na wejściach badanego układu w czasie występowania „okna”.

Długość cyklu testowania określają zbocza zewnętrznych sygnałów START/STOP. Interpretacja sygnatur jest bardzo prosta — albo jest ona prawidłowa (odpowiada sygnaturze otrzymanej teoretycznie lub eksperymentalnie, na wzorcowych US), albo nieprawidłowa. Stwierdzenie tego faktu nie wymaga od operatora specjalnych kwalifikacji (rys. 6).

Istotnym zagadnieniem w opisanej metodzie jest hierarchia przyłączenia poszczególnych wyjść licznika do wejść badanego układu. W przypadku czysto kombinacyjnych US jest ona dowolna. Dla bardziej skomplikowanych układów logicznych (przerzutniki, liczniki, rejestry) należy przeanalizować możliwość pojawienia się na wejściach kombinacji zabronionych. Tego typu wejścia (np. wejście zerujące i wpisujące w przerzutnikach) są połączone z licznikiem dwójkowym poprzez specjalne układy logiczne, wykluczające zabronione kombinacje testowe (np. deszyfratory).

Sygnaty podawane na wejścia testowanego układu mogą przyjmować znaczenia imitujące najgorsze wartości napięć: max. „0”, min. „1”. Na jego wyjścia można przyłączyć układy modelujące maksymalne obciążenie. Jeśli parametry statyczne lub dynamiczne układu scalonego nie mieszczą się w granicach tolerancji, to słowo wyjściowe miało inne znaczenie i tym samym sygnatury wyjść przyjmą nieprawidłowe znaczenia.

Opisany przyrząd pozwala na parametryczno-funkcjonalne testowanie, nie wymagające przeprowadzenia bezpośrednich pomiarów elektrycznych.

Należy oczekiwać, że zastosowanie analizy sygnatur w celach diagnostyki systemów cyfrowych, a zwłaszcza mikroprocesorowych, będzie coraz powszechniejsze. Świadczy o tym coraz częściej pojawiające się doniesienia o produkcji analizatorów sygnatur i włączaniu ich w zestawy pomiarowe przez tak renomowane firmy, jak HEWLETT-PACKARD, MILLENIUM czy TEKTRONIX [3].

#### LITERATURA

- [1] Die Signatur — Analyse. Elektronik, nr 1/1979.
- [2] Digital — Signaturanalyse, eine interessante Servicemethode für Digital — und Mikroprocessor — Systeme. Elektronik — Praxis, nr 11/1977
- [3] Hewlett-Packard. Application Note, 222-1, 1980
- [4] Hewlett-Packard Journal, May, 1977
- [5] Signature analysis-enhancing the serviceability of microprocessorbased industrial products. IECEI, 1978.

## INFORMATYKA o mikroprocesorach

Zgodnie z wcześniejszą zapowiedzią, w numerze 1/83 rozpoczniemy przegląd techniki mikroprocesorowej w Polsce i opublikujemy w nim następujące artykuły na ten temat:

- Sinkiewicz T.: Mikroprocesorowy system wspomagania projektowania
- Pluta A., Poznański Z., Szyńska J.: Wybrane systemy wspomagania dla mikrokomputerów rodziny MCS-48
- Pluszczok K.: RTDS-8 — system wspomagający uruchamianie systemów mikroprocesorowych
- Górnicki T., Szeżyńska M.: Środki wspomagające przygotowanie i uruchamianie oprogramowania mikrokomputerów jednoukładowych MCS-48
- Dańda J.: Oprogramowanie systemów mikroprocesorowych — praktyczne aspekty polityki producenta półprzewodników
- Jankowski M. T.: KRABUS — sprzęg wewnętrzno-kasetowy dla systemów mikrokomputerowych.

W następnych numerach zamierzamy rozszerzyć tę problematykę. Czytelników, którzy mogliby przekazać nam interesujące, związane z nią informacje, prosimy o bezwzględny kontakt z nami. (Red.)



# Algorytmy sortowania bez porównań

Zagadnienie sortowania obiektów według ustalonego porządku było znane dużo wcześniej niż skonstruowano pierwsze maszyny zdolne do jego wykonania. Dopiero bowiem powstanie nowoczesnej informatyki spowodowało burzliwy rozwój algorytmów rozwiązujących ten problem. Podyktowane to zostało z jednej strony ogromnym zapotrzebowaniem na sortowanie w praktycznym programowaniu [3], z drugiej zaś tym, że problem ten stanowi doskonałą podstawę do wszelkiego rodzaju prac teoretycznych z dziedziny analizy algorytmów. Mamy tu więc do czynienia z dość rzadkim przypadkiem zgodności zainteresowań praktyków i teoretyków.

Poniższe opracowanie ma na celu zaprezentowanie ciekawej klasy algorytmów sortowania. Oczywiście, nie ma ono zastępować słynnej książki D. E. Knutha [3], której lektura wydaje się nieodzowną dla dogłębnego poznania problemu sortowania (jakkolwiek praca niniejsza oparta jest głównie na tej książce), ma natomiast ułatwić Czytelnikowi ogólną orientację w omawianej dziedzinie. Jeśli zaś komuś wystarczy praktyczna procedura umożliwiająca natychmiastowe zakodowanie jednego z omawianych tu algorytmów, to można zaproponować sięgnięcie do działu ALGORYTMY — do umieszczonego w niniejszym numerze (str. 29) artykułu „Sortowanie przez podział dystrybucyjny”.

## PODSTAWOWE OKREŚLENIA

Na potrzeby tego artykułu sortowaniem nazwano znalezienie takiej permutacji  $p$  ciągu elementów  $E_1, E_2, \dots, E_N$  zbioru liniowo uporządkowanego, aby powstał ciąg spełniający warunek  $E_{p(1)} \leq E_{p(2)} \leq \dots \leq E_{p(N)}$ . Oczywiście, jak to najczęściej zdarza się w praktyce, dopuszcza się, aby operacją porównania wyznaczająca liniowy porządek była określona tylko na pewnej części elementu zwanej jego kluczem (klucz elementu  $E_j$  będzie oznaczany przez  $K_j$ ).

W dalszej części pracy terminy element i jego klucz będą utożsamiane i używane wymiennie. Zdecydowana większość znanych obecnie algorytmów sortowania, zarówno najprostszych, jak i bardzo skomplikowanych, to algorytmy porównawcze — tzn. takie, w których położenie danego elementu w ciągu uporządkowanym określa się na podstawie bezpośredniego porównywania z innymi elementami. Znacznie mniej liczna jest klasa algorytmów sortujących bez porównań, gdzie o miejscach elementów decydują ich własne wartości; algorytmy te określa się nieraz mianem sortowania przez rozsiew lub sortowania dystrybucyjnego. Można tu wspom-

nieć, że ten sposób był stosowany już w latach dwudziestych obecnego wieku w elektrycznych urządzeniach do sortowania kart dziurkowanych, a więc znacznie wcześniej przed powstaniem maszyn cyfrowych.

Dodatkowo, przy podawaniu złożoności algorytmów posłużono się powszechnie znanymi pojęciami rzędu funkcji  $f$  (oznaczanego  $O(f)$ ) oraz średniego i pesymistycznego czasu działania [1]. Dla uniknięcia trudności w rozumieniu przedstawionych metod sortowania zrezygnowano z programów napisanych w języku formalnym na rzecz słownych opisów algorytmów w języku potocznym.

## RODZAJE SORTOWANIA BEZ PORÓWNAŃ

Generalną podstawą wszystkich algorytmów tej klasy jest wykorzystanie faktu, iż zazwyczaj zbiór, z którego pochodzą elementy ciągu sortowanego, ma pewne dodatkowe własności, nie określone w ogólnym sformułowaniu zadania porządkowania. Pozwala to na efektywne tworzenie podciągów  $S_1, \dots, S_k$  takich, w których każdy podciąg  $S_i (i=2, 3, \dots, k)$  zawiera elementy większe od elementów wszystkich podciągów  $S_1, \dots, S_{i-1}$ . Podciągi takie bywają zwane przez wielu autorów koszykami, kulebkami czy kieszonkami, zaś od operacji „rozsypania” elementów do koszyków pochodzą nazwy dystrybucja lub rozsiew. Warto tu zauważyć, że taki sposób postępowania jest bliższy potocznemu, nieinformatycznemu znaczeniu słowa sortowanie — rozdzielanie wg gatunku, dzielenie na pewne grupy.

Zależnie od przyjęcia sposobu wyznaczania przynależności elementu do odpowiedniego podciągu, otrzymuje się inny rozsiew. Obecnie stosowane są trzy główne metody: zliczanie rozsiewu, rozsiew pozycyjny oraz rozsiew rangowy. Dalej przedstawiono algorytmy reprezentujące każdą z metod.

## SORTOWANIE PRZEZ ZLICZANIE ROZSIEWU

Algorytm zliczania rozsiewu (ang. *distribution counting*) daje się zastosować tam, gdzie znana jest z góry liczebność zbioru, z którego pochodzą elementy do posortowania. Istotne jest również, aby różnorodność elementów nie była zbyt wielka, tzn. by liczba różnych możliwych elementów była wyraźnie mniejsza od długości ciągu do posortowania. Dobrym modelem takiego przypadku są liczby całkowite z zakresu od  $p$  do  $r$  ( $p < r$ ), gdy  $r-p$  jest niezbyt duże w stosunku do  $N$  i dla niego zapisano algorytm Z. Oprócz tablicy elementów  $E_1, E_2, \dots, E_N$  w algorytmie wystąpi tablica liczników  $LICZ[p], \dots, LICZ[r]$ .

## Algorytm Z

- Z1. Wyzeruj liczniki  $LICZ[p]$  do  $LICZ[r]$
- Z2. Wykonaj krok Z3 dla  $j=1, \dots, N$ ; następnie przejdź do Z4.
- Z3. Zwiększ wartość  $LICZ[K_j]$  o 1.  
{ Po wykonaniu Z2 i Z3 każdy  $LICZ[i]$  zawiera liczbę kluczy o wartości równej  $i$  }
- Z4. Wykonaj krok Z5 dla  $i=p+1, \dots, r$ ; następnie przejdź do Z6.
- Z5. Nadaj  $LICZ[i]$  wartość  $LICZ[i]+LICZ[i-1]$ .  
{ Po wykonaniu Z4 i Z5 każdy  $LICZ[i]$  zawiera liczbę kluczy mniejszych lub równych  $i$ ; w szczególności  $LICZ[r]=N$  }
- Z6. Nadaj  $j$  wartość  $N$ .
- Z7. Jeśli  $j=0$ , to zakończ działanie. W przeciwnym przypadku, jeśli  $LICZ[K_j] < j$  — nadaj  $j$  wartość  $j-1$  i powtórz krok Z7;



Mgr MAREK SOBCZYK ukończył w 1980 r. studia o kierunku — oprogramowanie i metody informatyki na Wydziale Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego. Pracuje w Instytucie Organizacji Przemysłu Maszynowego, początkowo w Zakładzie Systemów Informatycznych, a obecnie w Zespole Kierowania Zastosowaniami Technik Komputerowych.



Jeśli  $LICZ[K_j] = j$  — zmniejsz  $LICZ[K_j]$  oraz  $j$  o 1 i powtórz krok Z7.

Jeżeli natomiast  $LICZ[K_j] > j$  nadaj  $T$  wartość  $E_j$ , i wartość  $LICZ[K_j]$ ,  $LICZ[K_j]$  wartość  $i-1$ .

Z8. Nadaj  $S$  wartość  $E_i$ ,  $k$  wartość  $LICZ[K_i]$ ,  $LICZ[K_i]$  wartość  $k-1$ ,  $E_i$  wartość  $T$ ,  $T$  wartość  $S$ , i wartość  $k$ . Jeśli  $i \neq j$ , to powtórz krok Z8; jeśli  $i = j$  nadaj  $E_i$  wartość  $T$ ,  $j$  wartość  $j-1$  i wróć do kroku Z7.

Kroki Z2 i Z3 wykonywane są  $N$  razy i powodują zliczanie liczby kluczy o każdej z wartości z zakresu od  $p$  do  $r$ . Następnie  $r-p$  razy „poprawia się” wartości liczników — tak, aby każdy  $LICZ[i]$  wskazywał ilość elementów mniejszych lub równych  $i$ . Kroki Z6 do Z8 stanowią z kolei właściwe porządkowanie tablicy  $E$  na podstawie informacji zawartych w  $LICZ[p] \dots LICZ[r]$ . Zastosowana tu metoda pozwala uniknąć tworzenia pełnej kopii tablicy  $E$ . Bliższe informacje o tej metodzie można znaleźć w [3]. Liczba operacji potrzebnych do uporządkowania elementów tym sposobem jest, podobnie jak w poprzednich krokach, proporcjonalna do  $N$  (dowód np. w [2]). W sumie więc, pamiętając o założeniu, że  $r-p \ll N$ , złożoność obliczeniowa całego algorytmu  $Z$  wynosi  $O(N)$ , natomiast wymagany obszar pamięci —  $N$  elementów oraz  $r-p$  liczników.

Przedstawiona tu metoda jest bardzo szybka, oczywiście przy zachowaniu podanych na wstępie ograniczeń. Choć na pierwszy rzut oka wydają się one bardzo ostre, to jednak algorytm  $Z$  daje się zastosować w wielu przypadkach, np. w celu częściowego posortowania ciągu według fragmentu klucza. Procedurę sortowania opartą o podaną zasadę (w języku PASCAL) można znaleźć w [1].

## SORTOWANIE PRZEZ ROZSIEW POZYCYJNY

Na potrzeby tej metody sortowania przyjmujemy, że klucz każdego elementu  $E_i$  zbudowany jest z  $k$  uporządkowanych znaków  $(c_k, \dots, c_2, c_1)$  gdzie  $0 \leq c_1 < M$ . Kolejność takich kluczy jest określona przez tzw. porządek leksykograficzny:

$$(c_k, \dots, c_2, c_1) < (d_k, \dots, d_2, d_1)$$

wtedy i tylko wtedy, gdy dla pewnego  $j$  ( $1 \leq j \leq k$ )  $c_j = d_j$  dla wszystkich  $i > j$ , ale  $c_j < d_j$ . Oczywiście, gdy  $j=0$ , to wtedy klucze są równe. Tak zdefiniowane klucze mogą być traktowane zarówno jako słowa długości  $k$  nad  $M$  elementowym alfabetem, jak i jako liczby zapisane w notacji pozycyjnej o podstawie  $M: c_k M^{k-1} + \dots + c_2 M + c_1$ .

Do zapisania algorytmu sortowania przez rozsiew pozycyjny przyjęto, że istnieje kolejka główna KOLEJ oraz kolejki pomocnicze  $Q[1], \dots, Q[M]$ . Symbol  $K_{ij}$  będzie oznaczał  $j$ -ty znak  $i$ -tego klucza.

## Algorytm P

- P1. Umieść  $E_1, E_2, \dots, E_N$  w KOLEJ.
- P2. Wykonaj kroki P3 do P6 dla  $j = 1, 2, \dots, K$ ; następnie zakończ działanie.
- P3. Opróżnij kolejki  $Q[1], \dots, Q[M]$ .
- P4. Dopóki kolejka KOLEJ niepusta, wykonuj krok P5; następnie przejdź do kroku P6.
- P5. Niech  $E_i$  będzie pierwszym elementem z KOLEJ. Przenieś  $E_i$  do kolejki  $Q[K_{ij}]$  usuwając go z KOLEJ.
- P6. Dołącz zawartość kolejek  $Q[1], \dots, Q[M]$  do KOLEJ.  
(Kolejka KOLEJ zawiera uporządkowane elementy  $E_i$ )

Algorytm  $P$  jest pewnego rodzaju uogólnieniem zasady wykorzystanej w algorytmie  $Z$  (choćby algorytmu  $Z$  nie można zastosować bezpośrednio w podanej wersji, gdyż każdy krok sortowania według kolejnego znaku musi mieć własność stabilności, tzn. nie może zmieniać uporządkowania elementów równych). W danym kroku ciąg wejściowy jest sortowany wg  $j$ -tego znaku metodą rozkładania do właściwych koszyków-kolejek, przy zachowaniu uporządkowania według znaków  $j-1, j-2, \dots, 1$ .

Przedstawiona tu zasada może mieć wiele praktycznych realizacji — zarówno ze względu na implementację kolejek, jak i na sposób podziału klucza na znaki. Na przykład — dla maszyn cyfrowych o reprezentacji

binarnej możemy przyjąć alfabet  $\{0,1\}$ , dla kluczy będących napisami — zwykły alfabet znakowy itd. Można również, na potrzeby konkretnego zastosowania, przyjąć pewien sztuczny podział na znaki (np. liczby dziesiętne traktować jako zapisane przy podstawie 100, sortując w jednym przebiegu według dwu cyfr). Oczywiście wraz ze wzrostem podstawy  $M$  będzie malała liczba koniecznych przebiegów algorytmu ( $k$ ), ale jednocześnie wzrośnie liczba kolejek  $Q$  (a więc wymagany obszar pamięci).

Wykonanie algorytmu  $P$  kosztuje  $kN$  umieszczeń elementów w kolejkach i  $kM$  połączeń kolejek. Jak więc widać, praktyczny czas realizacji algorytmu w dużej mierze zależy od tego, jak zostały zaprogramowane operacje kolejkowe (to one tu dominują), szczególnie — czy uniknięto fizycznego przenoszenia sortowanych elementów. Sortowanie przez rozsiew pozycyjny jest opłacalne (podobnie jak algorytm  $Z$ ) w przypadku dużej liczby elementów i niezbyt dużej długości kluczy.

Różne warianty algorytmów i programów opartych o metodę rozsiewu pozycyjnego przedstawione są w [3].

## SORTOWANIE PRZEZ ROZSIEW RANGOWY

Ostatnia metoda odnosi się do kluczy traktowanych jako liczby rzeczywiste z pewnego przedziału  $\langle \min, \max \rangle$ . Choćby tym razem zbiór możliwych wartości kluczy jest potencjalnie nieskończony, to jednak w dalszym ciągu metoda rozsiewu pozostaje skuteczna. Należy, oczywiście dokonać pewnych modyfikacji dotychczas przedstawionych sposobów. Ponieważ nie da się już dużej wyznaczyć odpowiedniego koszyka bezpośrednio na podstawie wartości klucza — należy kluczom przypisać pewne rangi. Zbiór możliwych rang jest skończony i jednakową rangę mogą otrzymać klucze o różnej wartości. Elementy o jednakowej randze będą rzecz jasna trafiać do jednego koszyka.

Naturalnym sposobem tworzenia takich rang jest podzielenie przedziału  $\langle \min, \max \rangle$  na  $M$  ( $M \leq N$ ) podprzedziałów jednakowej długości  $h = (\max - \min)/M$ . Otrzymuje się zatem dla  $j = 0, 1, \dots, M-2$  podprzedziały lewostronnie domknięte  $\langle \min + jh, \min + (j+1)h \rangle$  oraz podprzedział domknięty  $\langle \min + (M-1)h, \max \rangle$ . Każdy element ma więc rangę  $j = \text{ENTIER}((K_i - \min)/h)$  i będzie wstawiony do koszyka numer  $j+1$  z wyjątkiem przypadku  $E_i = \max$ , gdy numer ten pozostanie  $M$ .

Przedstawiony poniżej algorytm (tzw. algorytm sortowania na wielu listach) jako koszyki stosuje listy  $L_1, L_2, \dots, L_M$ . Ponieważ elementy takiej listy nie muszą być równe, należy je uporządkować. Najlepiej zastosować w tym celu prostą metodę sortowania przez wstawianie na listach (ang. *list insertion*), omówioną np. w [3], tzn. praktycznie będzie to dołączanie elementu do listy — tak, aby była ona zawsze uporządkowana.

## Algorytm R

- R1. Nadaj  $h$  wartość  $(\max - \min)/M$ .
- R2. Dla każdego  $i = 1, 2, \dots, N$  wykonaj kroki R3 i R4; następnie przejdź do kroku R5.
- R3. Nadaj  $j$  wartość  $\text{ENTIER}((K_i - \min)/h)$ . Jeśli  $j < M$  to zwiększ  $j$  o 1.
- R4. Umieść (metodą list insertion) element  $E_i$  na liście  $L_j$ .
- R5. Dla każdego  $k = 2, 3, \dots, M$  dołącz  $L_k$  na koniec  $L_1$ .  
(Lista  $L_1$  zawiera uporządkowane elementy  $E_i$ )

Pesymistyczny czas działania algorytmu  $R$  jest  $O(N^2)$  — dzieje się tak, gdy w rzeczywistości tworzy się tylko jedna lista i wtedy działa wyłącznie *list insertion*. Natomiast przy założeniu, że rozkład danych w przedziale  $\langle \min, \max \rangle$  jest jednostajny, średni czas wynosi  $O(N^2/M)$  [1]. Także w [1] można znaleźć przykładowy program w PASCALU. Pewną ciekawą modyfikację algorytmu  $R$  przedstawiono w [2]: po pierwsze tworzy się  $N$  podprzedziałów ( $M=N$  co powoduje, że średni czas wynosi  $O(N)$ ), po drugie — dzięki znajdowaniu mediany (elementu środkowego) — czas pesymistyczny jest rzędu  $O(N \log N)$ . Praktyczną procedurę dla tego algorytmu (o nazwie DPS) zawiera wspomniany we wprowadzeniu artykuł z cyklu ALGORYTMY.



## PORÓWNANIE SZYBKOŚCI

Tym, których nie przekonują podane wyżej teoretyczne złożoności obliczeniowe, autor proponuje zapoznanie się z tablicami porównującymi szybkość działania różnych algorytmów. Tabela 1 (przytoczona za [2] zawiera wyniki testów empirycznych przeprowadzonych na maszynie CDC CYBER 73 dla algorytmu DPS oraz dla — u- chodzących za bardzo szybkie — algorytmów porównaw- czych QUICKSORT i HEAPSORT (czasy podano w mi- lisekundach). W tabeli 2 (wg [3]) podano czasy wyko- nania na hipotetycznej maszynie MIX dla algorytmów Z, P, R oraz QUICKSORT, HEAPSORT i metody pro- stego wyboru (tzn. sortowanie poprzez wybranie elemen- tu największego, drugiego co do wielkości (itd.), o zło-

Tabela 1

Algorytm	N				
	100	1000	5000	10000	15000
DPS	23	254	1202	2552	3807
QUICKSORT	25	351	2072	4482	6984
HEAPSORT	25	307	2079	4832	7721

żoności  $O(N^2)$ ). Tym razem przedstawiono czasy względne, w stosunku do przyjętego za jednostkę czasu wykonania QUICKSORTU dla  $N = 16$ .

Tabela 2

N	Algorytm					
	Z	P	R	QUICK- SORT	HEAP- SORT	Prosty wybór
16	21,3	8,7	1,3	1	2,2	1,8
1000	65,7	75,6	72,4	153,1	3281	5185,4

## LITERATURA

- [1] Banachowski L., Kreczmar A.: Elementy analizy algorytmów. Warszawa, WNT 1982.
- [2] Dobosiewicz W.: Sortowanie przez podział dystrybucyjny (praca doktorska). Warszawa, Uniwersytet Warszawski, 1979.
- [3] Knuth D. E.: The Art of Computer Programming, Vol. 3/Sorting and Searching, Reading, Mass. Addison-Wesley, 1973.

ANDRZEJ KUKUŁA, URSZULA SZMIDT  
Zakład Problemów Organizacji i Zarządzania PĀN  
Bytom

# Analiza przydatności USZBD RODAN dla baz danych o wysokiej dyspozycyjności

W latach 1980—1981 prowadzono badania nad przydatnością Uniwersalnego Systemu Zarządzania Bazy Danych RODAN — do zarządzania bazami danych w zintegrowanym systemie komputerowego zarządzania produkcją podstawową huty żelaza. Jednym z głównych wymogów stawianych tego rodzaju bazom jest wysoka dyspozycyjność.

Dyspozycyjność bazy danych związana jest z jej niezawodnością. Jeżeli baza danych skonstruowana jest tak, że każde jej uszkodzenie, niezależnie od tego ile elementów obejmuje, wymaga wyłączenia jej z ruchu celem poprawienia błędów, to z punktu widzenia niezawodności jest ona urządzeniem dwustanowym naprawialnym. Dyspozycyjność bazy można wówczas rozumieć jako prawdopodobieństwo gotowości do pracy. Dla celów praktycznych prawdopodobieństwo to utożsamiane jest często ze współczynnikiem gotowości do pracy, tj. stosunkiem średniego czasu kolejnego uszkodzenia do średniego czasu między uszkodzeniami (obejmującego również czas naprawy).

Gdy zaś uszkodzenia lokalne bazy danych wymagają wyłączenia z ruchu jedynie elementów uszkodzonych, względnie najbliższego związanego z nimi otoczenia (sta-

nowiącego ciągle jeszcze tylko część systemu), wówczas baza jest urządzeniem wielostanowym naprawialnym. Miejsce współczynnika gotowości do pracy, określonego dla urządzeń dwustanowych, zajmuje w przypadku urządzeń wielostanowych wartość oczekiwana efektywności systemu.

Analizując uszkodzenia, na które narażona jest baza danych, a których przyczyną mogą być: źle działający system operacyjny, źle oprogramowanie narzędziowe czy programy użytkownika lub błędne dane — dojdziemy do wniosku, że w trakcie eksploatacji bazy będziemy mieć najczęściej do czynienia z uszkodzeniami lokalnymi. Ich liczba będzie względnie stała w części spowodowanej pracą sprzętu, systemu operacyjnego i oprogramowania narzędziowego. Będzie jednak rosła proporcjonalnie do liczby programów i danych aktualizujących zawartość bazy.

Ponieważ baza danych jest urządzeniem przeznaczonym dla wielu zastosowań, w przypadku jej intensywnej eksploatacji gotowość systemu do pracy może być znikoma. Dlatego też znacznie lepszą dyspozycyjność można uzyskać realizując bazę jako urządzenie wielostanowe na-



prawialne. Aby to osiągnąć trzeba jednak dokonać — w trakcie projektowania — podziału bazy na części rozłączne, (z punktu widzenia naprawy), oraz trzeba dysponować systemem zarządzania bazą, który wyposażony jest w odpowiednie narzędzia do ochrony danych.

## CHARAKTERYSTYKA BADANEJ BAZY

Badana baza przeznaczona jest dla komputerowego systemu zarządzania produkcją podstawową kombinatu metalurgicznego. Ma ona zaspokoić potrzeby informacyjne trzech podsystemów: technicznego przygotowania, planowania oraz rozliczania produkcji i zbytu. Technologicznie baza jest rozwiązana w oparciu o system zarządzania zrealizowany wg zaleceń zawartych w raportach komitetu CODASYL.

W schemacie bazy zdefiniowano 66 typów rekordów. Są one rozmieszczone w 14 obszarach, przy czym niektóre typy rekordów występują tylko w jednym obszarze, zaś inne — w wielu. W schemacie zdefiniowano również 48 typów dwupoziomowych struktur hierarchicznych (setów), które odwzorowują ważniejsze powiązania występujące między przedmiotami zarządzania. Elementy niektórych tego rodzaju struktur zawarte są całkowicie w ramach tylko jednego obszaru. Generalnie z tego punktu widzenia obszary można podzielić na dwa rodzaje:

**obszary samodzielne**, w projektowanej bazie są to np. obszary zawierające rekordy opisujące obiekty przynależne bezpośrednio do samodzielnych organizacyjnie wydziałów produkcyjnych

**obszary powiązane**, czyli zawierające rekordy opisujące obiekty lub zdarzenia, dotyczące więcej niż jednego wydziału i wchodzące w skład struktur (setów) obejmujących więcej niż jeden obszar.

Zakłada się, że w docelowej bazie danych będzie ok. 1 mln wystąpień rekordów. Liczba wystąpień zarówno w ramach typów rekordów, typów setów, jak i rekordów składowych w wystąpieniach setów tego samego typu jest bardzo zróżnicowana.

Szacuje się, że jedynie 20% wystąpień rekordów będzie posiadało względnie stałe wartości, natomiast 80% wystąpień rekordów będzie — w skali roku — podlegało wymianie. Zdarzenia powodujące konieczność wymiany lub modyfikacji wartości wystąpień rekordów bazy występują w zakresie technicznego przygotowania i planowania produkcji co dobę, natomiast w zakresie sterowania i kontroli realizacji produkcji i zbytu — na każdej z trzech zmian produkcyjnych. Ponadto w różnych miejscach obszaru zarządzania może wystąpić więcej niż jedno zdarzenie równocześnie.

Baza danych i system informatyczny mają być zrealizowane technicznie na komputerze R-32, wyposażonym w pamięć operacyjną o wielkości 1 MB, dyski magnetyczne o pojemności 30 MB, sieć monitorów ekranowych oraz inne standardowe urządzenia wejścia-wyjścia.

Spodziewana zatem intensywne eksploatacja stosunkowo dużej bazy danych (zajmującej kilka pakietów dyskowych o pojemności 30 MB) za pomocą sprzętu o niskich parametrach niezawodnościowych [2] nakazywała poszukiwanie możliwości rozwiązania bazy jako systemu wielostanowego naprawialnego.

## ANALIZA NARZĘDZI USZBD RODAN DO OCHRONY DANYCH

RODAN jest najbardziej uniwersalnym, dostępnym w kraju do eksploatacji na komputerze R-32 systemem zarządzania bazą danych.

Spośród różnorodnych narzędzi do ochrony danych jakimi dysponują współczesne SZBD, dla osiągnięcia powyżej scharakteryzowanego celu, system zarządzania bazą danych winien posiadać następujące narzędzia:

- do bezpiecznego przetwarzania transakcji w wielozadaniowym trybie pracy
- do selektywnego kopiowania, wyłączenia z pracy i naprawy samodzielnych obszarów bazy w trakcie pracy systemu. Roważymy kolejno powyższe zagadnienia.

**Bezpieczne przetwarzanie transakcji** związane jest z pojęciem zwartej sekwencji akcji, która ma tę własność, że baza danych jest w stanie zgodnym (jest spójna) przez i po jej wykonaniu, nie musi natomiast być zgodna w trakcie wykonywania. Dlatego SZBD powinien sprawdzić poprawność wykonania takiej sekwencji, a w razie stwierdzenia nieprawidłowości — odrzucić wykonanie transakcji i automatycznie odwrócić ewentualne, już dokonane w ramach owej sekwencji, zmiany w bazie. Obecnie RODAN zawiera mechanizm pozwalający na odwrócenie zmian spowodowanych błędnie działającą komendą języka manipulacji danymi. Autorzy RODANU zapowiedzieli także wprowadzenie mechanizmu pozwalającego na bezpieczną realizację zwartej sekwencji akcji składającej się z wielu komend [3].

**Ochrona istnienia danych.** Narzędzia do selektywnego kopiowania, wyłączenia z pracy i naprawy samodzielnych obszarów bazy w trakcie pracy systemu należą do grupy narzędzi do ochrony istnienia danych. RODAN posiada obecnie trzy narzędzia tego typu: mechanizm umożliwiający statyczne kopiowanie oraz odtwarzanie z kopii bazy danych, mechanizm dynamicznego rejestrowania zmian realizowanych w bazie — w postaci kopii stron przed i po zmianie — oraz narzędzia do statycznego odtwarzania z kopii bazy danych.

Narzędzia te wykorzystano w praktyce. Doświadczenia w tym względzie są następujące:

• Autorzy RODANU polecają — jako narzędzie do statycznego kopiowania bazy — standardowy program OS/JS o nazwie IEHDASDR. Program ten działa poprawnie pod warunkiem, że tworzona kopia nie przekracza pojemności jednej szpuli taśmy magnetycznej. W przeciwnym przypadku, kopii nie można przepisać na dysk. Program ten ma zastosowanie przy tworzeniu kopii całej bazy. Do tworzenia indywidualnych kopii poszczególnych obszarów można zastosować standardowy program OS/JS o nazwie IEHMOVE.

• Rejestracja kopii zmienianych stron bazy danych przebiega poprawnie i bez zakłóceń, jak również poprawne jest odtworzenie bazy danych przy zastosowaniu tego rejestru. Odtwarzaniu podlegają wybrane i określone dla programu obszary bazy danych. Jest to zatem narzędzie do selektywnego odtwarzania bazy danych. Niestety, jak już podano wcześniej, odtwarzanie jest statyczne, czyli żadne programy aplikacyjne nie mają w tym czasie dostępu do całej bazy.

\* \* \*

W zakresie bezpiecznego przetwarzania transakcji, po uzupełnieniu mechanizmów realizacji zwartej sekwencji akcji (jak to opisano wcześniej) RODAN zapewnia całkowicie możliwość funkcjonowania bazy danych jako systemu wielostanowego naprawialnego. W zakresie ochrony istnienia danych, w porównaniu z innymi systemami zarządzania (np. [1]), RODAN tylko częściowo zapewnia wykorzystanie przewagi, jaką mają — z punktu widzenia dyspozycyjności — systemy wielostanowe nad dwustanowymi. Projektant bazy może bowiem uwzględnić przy jej podziale na obszary, charakterystyki niezawodnościowe przyszłego systemu i dążyć do minimalizacji czasu naprawy uszkodzeń lokalnych. RODAN nie umożliwia natomiast czasowego wyłączenia i naprawy uszkodzonych elementów (obszarów bazy) przy zapewnieniu ciągłej pracy systemu, z jedynie obniżoną w czasie owych połączeń efektywnością.

Wiele współczesnych SZBD [5] dysponuje specjalnym narzędziem dla maksymalnego skrócenia czasu naprawy niektórych uszkodzeń. Narzędzia te stanowią tzw. zbiory



podwójne. Wszystkie dane bazy są wówczas utrzymywane na dwu niezależnych nośnikach fizycznych. Jest to narzędzie kosztowne w eksploatacji, ponieważ wymaga poważnych dodatkowych środków w postaci pamięci dyskowych i czasu przetwarzania. Niemniej gdy wymagania odnośnie dyspozycyjności bazy są bardzo wysokie — jest ono stosowane.

Autorzy RODANU zdają sobie sprawę ze słabości niektórych obecnych narzędzi. Już bowiem z początkiem 1980 roku zaanonsowali wprowadzanie nowego systemu ochrony danych [3].

## LITERATURA

- [1] DMS 1100, Level 7R1, System Support Functions, UNIVAC, 1978
- [2] Drewniak W.: Niezawodność systemów komputerowych ODRA 1305 i R-32. Informatyka nr 7/1980
- [3] Stramowski A.: Odtwarzanie stanów bazy danych. Nowości Informatyki, CPIZI, Warszawa, 1980
- [4] Sztarski M.: Niezawodność i eksploatacja urządzeń elektronicznych. WKiŁ, Warszawa, 1972
- [5] TIP — Transaction Interface Package 1100, UNIVAC, 1978
- [6] Uniwersalny System Zarządzania Bazą Danych RODAN. Dokumentacja użytkowa, CPIZI, Warszawa, 1979.

## KONFERENCJE

### SPIS'82

Kolejne, szóste seminarium SPIS'82 odbyło się w dniach 13—15 października 1982 r. w Jachrance k. Warszawy pod hasłem „Systemy informatyczne w warunkach reformy gospodarczej”. Podobnie jak w latach ubiegłych, organizatorami Seminarium były: Sekcja Organizacji i Przetwarzania Danych Komitetu Statystyki i Ekonometrii PAN, Komisja Informatyki PTE oraz Ośrodek Badawczo-Rozwojowy SPIS i Zarząd Mechanizacji i Automatykacji Opracowań Statystycznych GUS. Program Seminarium obejmował obrady plenarne, obrady w dwóch sekcjach oraz dyskusję panelową.

Podczas obrad plenarnych w pierwszym i ostatnim dniu Seminarium wygłoszono referaty główne, spośród których należy wyróżnić wystąpienia prof. T. Walczaka „Zmiany w sprawozdawczości statystycznej kształtowane reformą gospodarczą”, doc. J. Kisielnickiego „Warunki wspomaganie reformy gospodarczej przez systemy informatyczne” oraz prof. Peche „Koncepcja Systemu Informatycznego Rachunkowości a reforma gospodarcza”. Zmieniające się pod wpływem reformy gospodarczej potrzeby informacyjne powodują, że szereg elementów systemu SPIS wymaga zmian dotyczących sposobu udostępniania wyników i współpracy z odbiorcami informacji, a także treści i obiegu informacji. Obecnie rozszerzono nie tylko zakres dostępności informacji statystycznych, ale również dostępu: obok form tradycyjnych (publikacje) GUS udostępnia informacje zapisane na taśmach magnetycznych odbiorcom posiadającym komputery. Następuje sukcesywne doskonalenie informacji statystycznej w kierunku lepszego dostosowania tej treści do zmienionych potrzeb odbiorców. Trzeci kierunek zmian, to zastąpienie resortowego obiegu informacji przez obieg terenowy (przedsiębiorstwa — wojewódzkie urzędy statystyczne — GUS). Rozszerzenie terenowego obiegu informacji to najpilniejsze, a zarazem najtrudniejsze zadanie statystyki państwowej.

Niestety, w założeniach reformy gospodarczej prawie nie mówi się o systemach informatycznych, a więc o zarządzaniu wspomaganym techniką komputerową. Na tym tle alarmujący jest fakt, że jesteśmy krajem, gdzie obecne dostawy nowego sprzętu nie pokrywają nawet naturalnych ubytków wynikających z fizycznego zużycia komputerów.

W licznych referatach postulowano konieczność dokonania znacznego postępu w jakości projektowania systemów informatycznych. Konieczne są też modyfikacje systemów już istniejących, w tym również systemów rządowych. Jeden z autorów stwierdził nawet, że „powodzenie reformy gospodarczej w większym stopniu niż przy dotychczasowym systemie zarządzania zależy będzie od dobrze zaprojektowanego i sprawnie działającego systemu informacyjnego wspartego techniką komputerową...”.

Reforma gospodarcza zmienia nie tylko warunki, ale również narzędzia zarządzania, do których należy zaliczyć System Informatyczny Rachunkowości (SIR). U uruchomienie prototypu SIR nastąpi w 1983 r., natomiast do końca 1985 r. ma być on wprowadzony do eksploatacji w większej liczbie różnych przedsiębiorstw przemysłowych. Za pilnością jego wdrożenia przemawia przy tym pogłębiający się deficyt kadr w planach finansowo-księgowych.

W sekcji I referaty przygotowano pod kątem następujących tematów: „Kierunki rozwoju SPIS w warunkach reformy gospodarczej” oraz „Systemy informatyczne planowania centralnego w warunkach reformy”. Podczas dyskusji w tej sekcji ciekawym, choć — moim zdaniem — tylko częściowo słusznym stwierdzeniem było ostrzeżenie, że skutkiem ograniczeń w projektowaniu i wdrażaniu nowych systemów informatycznych, następować będzie odrażanie się starych struktur zarządzania w wyniku braku szybkiej informacji bieżącej.

Sekcja II, tematycznie nieco bogatsza, obejmowała referaty ukierunkowane na trzy tematy: „Metodyczne problemy rozwoju centralnych systemów informatycznych w warunkach reformy gospodarczej”, „Branżowe i resortowe systemy informatyczne a reforma gospodarcza” oraz „Informacja naukowo-techniczna jako czynnik wdrażania reformy gospodarczej”.

Z obszernej dyskusji jaka toczyła się podczas obrad tej sekcji, chciałbym wyeksponować kilka zagadnień, w pierwszym rzędzie — permanentnego braku w kraju efektywnej informacji o informacjach. Cenną inicjatywą w tym zakresie jest przystąpienie do opracowania tak potrzebnego ogólnokrajowego katalogu produktów programowych i systemów informatycznych. Innym szczególnie ważnym zagadnieniem jest konieczność coraz szerszego korzystania z gotowego oprogramowania, które najskuteczniej podnosi efektywność stosowania informatyki. W tym kontekście apelowano o przełamywanie barier psychologicznych, jakie występują przy wykorzystywaniu obcego oprogramowania.

W dyskusji panelowej, odbywającej się pod hasłem „SPIS a resortowe i branżowe systemy informatyczne w warunkach reformy gospodarczej”, oprócz podstawowego tematu podejmowano zagadnienia dotyczące samych systemów resortowych, a zwłaszcza ich zakresu informacyjnego. Wskazywano na przypadki opracowywania i rozsyłania przez ministerstwa ankiet jako metody gromadzenia informacji o potrzebnej tematyce. Część tej dyskusji poświęcono źródłom i pomiarowi informacji oraz możliwości stwierdzenia ich wiarygodności.

Z drugiej części obrad plenarnych chciałbym zasygnalizować referat J. Iszkowskiego „Jak zaspokoić potrzeby informacyjne terenowych organów władzy?”, zawierający analizę istniejących już systemów informatycznych dla wojewódzkich organów władzy z punktu widzenia stopnia zaspokożenia ich potrzeb informacyjnych.

W końcowej dyskusji Seminarium wypuklono znaczenie Wojewódzkich Banków Danych (WBD), które powinny szybko nadążać za zmieniającymi się gospodarczymi i politycznymi potrzebami informacyjnymi województwa. WBD powinny stać się podstawowym narzędziem informacyjnym władz terenowych.

W podsumowaniu obrad stwierdzono potrzebę szerszego udziału w seminarium SPIS pracowników nauki, a jednocześnie zwiększenia reprezentacji przedstawicieli odbiorców informacji dostarczanych przez system SPIS.

A. SOKOŁOWSKI



## Programowany rejestrator PRS-4 — podstawa automatyzacji kopalni węgla kamiennego

Prawidłowe prowadzenie procesów technologicznych i kontrola stanu bezpieczeństwa kopalni wymagają ciągłej kontroli wielu parametrów. Liczba sygnałów — doprowadzanych do centrum dyspozytorskiego w przeciętnej kopalni — sięga tysiąca, a wielkość tę przekracza w kopalniach dużych bądź w kopalniach o znacznym zagrożeniu wybuchem metanu. Ciągła kontrola tak dużej liczby parametrów jest niemożliwa bez zastosowania techniki cyfrowej, która na bieżąco analizuje stan obiektu, przetwarza uzyskiwaną informację wg żądanych algorytmów oraz informuje kierownictwo kopalni o istotnych zdarzeniach występujących w podległym mu obiekcie. Rozwój urządzeń i systemów dyspozytorskich w ostatnich 20 latach w górnictwie przebiegał wolno — od prostych dyspozytorni wyposażonych jedynie w niedoskonałe środki łączności, aż po komputerowe systemy nadzoru i kontroli. Technika cyfrowa była wprowadzana w kopalniach ostrożnie, a uzyskiwane efekty nie zawsze potwierdzały przyjęte założenia. Negatywne wyniki były najczęściej spowodowane dużą zawodnością sprzętu i niewłaściwą koncepcją systemów.

Od początku stosowania maszyn cyfrowych w górnictwie dały się zauważyć dwie różne tendencje:

- stosowanie jednej maszyny cyfrowej, której można powierzyć nadzór i przetwarzanie informacji dotyczących wszystkich procesów występujących w kopalni
- stosowanie małych systemów cyfrowych, dla rozwiązania osobno każdego zagadnienia (procesu).

Druga tendencja zakłada stosowanie tylu odrębnych zestawów cyfrowych (wyposażonych w czujniki i systemy transmisji), ile w danej kopalni występuje skomplikowanych problemów wymagających zastosowania takiego sprzętu. Ze względu na łatwość rozpowszechnienia tak opracowanych systemów, jak również na większą ich niezawodność — w Polsce przyjęto tę drugą drogę: stosowanie zestawów cyfrowych — modułów. System dyspozytorski oparty na zestawie modułów nazwano dyspozytornią modułową.

Niezaprzeczalną zaletą przyjętej koncepcji jest:

- łatwość wdrażania, wynikająca z ograniczonego zasięgu oddziaływania jednego modułu
- łatwość dostosowania algorytmu działania do zmieniających się wymagań użytkownika
- większa niezawodność eksploatacyjna, wynikająca z podziału funkcji.

\*

Prace nad zastosowaniem informatyki do kontroli i sterowania procesów technologicznych, występujących w polskich kopalniach, zostały rozpoczęte na przełomie lat 1969—1970 w ówczesnych Zakładach Konstrukcyjno-Mechanicznych Przemysłu Węglowego (ZKMPW). Efektem tych prac było eksperymentalne wdrożenie w 1970 r. — w zautomatyzowanej doświadczalnej kopalni „Jan” — systemu kompleksowej automatyzacji „S”, wykorzystującego minikomputer MKJ-25 z pamięcią buforową jako kanałem przemysłowym. W wyniku uzyskanych doświadczeń powstał prototypowy system automatyzacji, wdrożony na przełomie 1973/1974 w jednej z kopalni Jaworznioko-Mikolowskiego Zjednoczenia PW.

W systemie wykorzystano ten sam zmodyfikowany minikomputer z nowo opracowanym kanałem przemysłowym UZO-4 (patent PRL nr 89 299). Zastosowany minikomputer charakteryzował się przestarzałą architekturą, małą szybkością działania i skąpym — opracowanym w całości w ZKMPW — oprogramowaniem (prosty assembler, programy testujące i edycyjne).

Doświadczenie zdobyte dzięki dwóm pierwszym instalacjom wykazało, że w celu szerokiego rozpowszechnienia systemów kontroli i sterowania w górnictwie konieczne jest wykorzystanie prostego, taniego i niezawodnego minikomputera z bogatym oprogramowaniem podstawowym i systemowym. Przemysł krajowy nie dysponował takim minikomputerem, a import nie wchodził w rachubę, zdecydowano się zatem na takie zaprojektowanie sprzętu komputerowego, aby akceptował on w pełni oprogramowanie podstawowe i systemowe jednego z popularnych w świecie minikomputerów. Pozwoliło to na znaczne skrócenie czasu realizacji przedsięwzięcia. Dostępne źródła literaturowe wskazują, że opracowanie sprzętu pochłania jedynie 30%, podczas gdy nakłady na opracowanie oprogramowania podstawowego i systemowego wynoszą ok. 70%.

W wyniku prac konstrukcyjnych, w końcu 1974 r. powstał prototyp minikomputera SMC-3. Wyprodukowano 17 egzemplarzy tego minikomputera, przy czym połowa z nich została wdrożona w górnictwie i hutnictwie. Pozostałe wykorzystano w laboratoriach oraz pracach inżynierijno-technicznych.

Zbrane w ciągu kilku lat eksploatacji doświadczenia, a także rozwój krajowej bazy elementowej — pozwoliły na zaprojektowanie przemysłowej wersji minikomputera, nazwanej rejestratorem PRS-4.

Prototyp tego minikomputera wykonano w 1978 roku, a seryjną produkcję rozpoczęto w 1980. Liczba wyprodukowanych egzemplarzy rejestratora PRS-4 przekroczyła obecnie 50 sztuk. Minikomputer ten stanowi jednostkę centralną systemów kontroli i nadzoru dyspozytorskiego w kopalniach, opracowanych w OBR SMEAG. Niezależnie od zastosowań górniczych, wykorzystuje się go z powodzeniem w innych gałęziach przemysłu (hutnictwo, energetyka, kolejnictwo), a także w laboratoriach wyższych uczelni i instytutów.

### CHARAKTERYSTYKA OGÓLNA REJESTRATORA

Rejestrator PRS-4 jest uniwersalnym, modułowym minikomputerem o stałej liście rozkazów. Składa się on z 16-bitowego procesora, pamięci operacyjnej i kanału wejścia-wyjścia. W kanale we-wy pracują jednostki sterujące urządzeniami peryferyjnymi i pamięciami zewnętrznymi oraz układy kanału przemysłowego. Konfiguracja jednostek sterowania i układów kanału przemysłowego jest dowolna, co umożliwia łatwe dostosowanie struktury rejestratora do konkretnego obiektu i zadania. Wszystkie układy rejestratora PRS-4, w tym również jeden z typów pamięci operacyjnej, wykonane są w mechanicznym standardzie systemu CAMAC.

Rejestrator wyposażony jest standardowo w układ detekcji zaniku napięcia zasilania oraz w układ restartu, a opcjonalnie może być wyposażony w moduł PROPARG, zawierający układy protekcji pamięci (kontrola parzystości



i ochrona pamięci) oraz moduł DMA bezpośredniego dostępu do pamięci z układem „sprzętowego” wykrywania makroinstrukcji.

## Dane techniczne

### Procesor

Typ procesora: równoległy, jednoadresowy  
 Długość słowa: 16 bitów + 2 bity parzystości (opcja)  
 Liczba rejestrów roboczych: 7 (A, B, M, T, P, E, O — w tym dwa równoprawne akumulatory A i B, którym przyporządkowano adresy 0 i 1 pamięci operacyjnej)  
 Liczba rozkazów: 45 (po jednym słowie); w tym : rozkazy adrese-wielofazowe, rozkazy rotacji i przesunięć rejestrów A, B i E — jednofazowe, rozkazy skoków warunkowych — jednofazowe, rozkazy wejścia-wyjścia — jednofazowe  
 Istnieje możliwość łączenia do ośmiu mikrorozkazów w jedno słowo rozkazowe, wykonywane w jednym cyklu maszynowym dla grupy rozkazów skoków warunkowych.  
 Adresowanie: bezpośrednie i pośrednie  
 Liczba adresowań pośrednich: nieograniczona  
 Czas realizacji rozkazu bezadresowego: 2 $\mu$ s  
 Minimalny czas realizacji rozkazu adresowego: 4  $\mu$ s

### Pamięć operacyjna

Rodzaj pamięci: ferrytowa, rdzeniowa z cyklem dzielnym  
 Minimalna pojemność: 4 K słów  
 Maksymalna pojemność: 32 K słów  
 Czas dostępu: nie większy niż 390 ns  
 Organizacja pamięci: stronicowa  
 Pojemność strony: 1 K słów  
 Ostatnie 64 komórki chronione dla przechowywania programu ładującego LOADER  
 Istnieje możliwość zastosowania wolniejszych pamięci, co jednak wiąże się z odpowiednim spowolnieniem rejestratora. Przykładowo — dla pamięci ferrytowej FJP 8/18/1 produkcji ELWRO z czasem dostępu 480 ns czas realizacji rozkazu bezadresowego wyniesie 2,33  $\mu$ s. Możliwe jest również zastosowanie pamięci półprzewodnikowej typu RAM (zachowanie restartu wymaga buforowania napięcia zasilania).

### Kanał wejścia-wyjścia

Liczba gniazd w kasiecie procesora dla modułów wejścia-wyjścia: 16  
 Możliwość rozszerzenia liczby gniazd przez zastosowanie multipleksera o 16 gniazdach dla każdego multipleksera  
 System przerwań: wektoryzowany  
 Kanał bezpośredniego dostępu do pamięci operacyjnej: opcja dla pierwszych ośmiu modułów w kanale  
 Liczba kodów selekcyjnych przerywających dla szczeliny: 2 (szczeliny 1–16)  
 Liczba kodów selekcyjnych nieprzerywających dla szczeliny: 3 (szczeliny 4–16)  
 Magistrala danych: 16-bitowa, dwukierunkowa.

## Struktura sprzętowa rejestratora

Podstawową strukturę rejestratora PRS-4 przedstawiono na rysunku 1. Można w niej wyróżnić cztery zasadnicze bloki:

- procesor wraz z opcjami (moduły DMA i PROPAR)
  - pamięć operacyjną
  - moduły sterowania urządzeniami peryferyjnymi i pamięciami zewnętrznymi
  - moduły kanału sprzężenia z obiektem.
- Wymienione bloki współpracują ze sobą za pośrednictwem dwóch niezależnych magistrali:
- magistrali pamięci łączącej pamięć operacyjną z procesorem
  - wspólnej magistrali we-wy, łączącej procesor z modułami sterowania i modułami kanału przemysłowego.

Rysunek 1 przedstawia także sposób powiązania procesora z urządzeniami peryferyjnymi i pamięciami zewnętrznymi, wskazując jednocześnie typy użytych modułów sterujących.

Jedną z własności rejestratora PRS-4 jest możliwość przyłączenia praktycznie wszystkich spotkanych w kraju urządzeń peryferyjnych za pośrednictwem tylko dwóch typów modułów: uniwersalnego dwukierunkowego rejestru 8-bitowego KI-420 oraz uniwersalnego sprzęgu szeregowego KI-430.

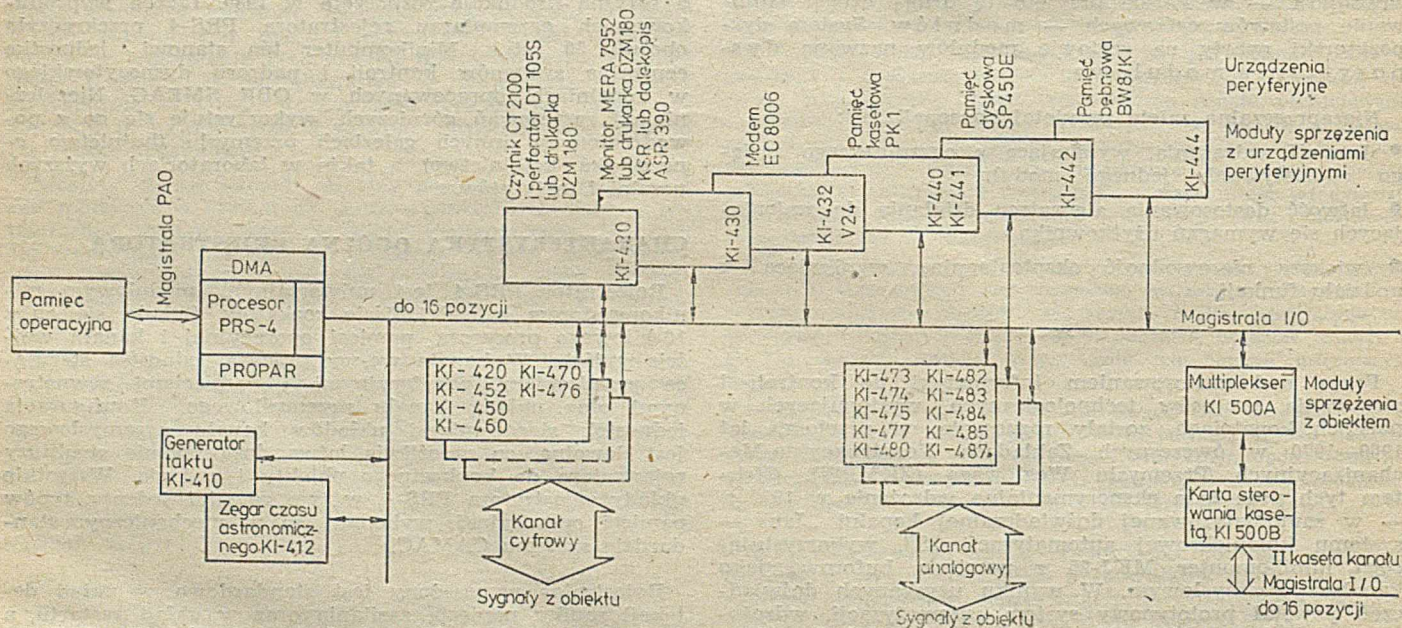
Niezależnie od wymienionych wyżej modułów sterujących — rejestrator posiada bogaty asortyment modułów kanału przemysłowego, umożliwiających sprzężenie z obiektem. W kanale przemysłowym można wyróżnić moduły specjalistyczne oraz moduły wprowadzania i wyprowadzania sygnałów cyfrowych i analogowych.

### Moduły specjalistyczne

- Programowany generator przerwań — KI-410
- Zegar czasu astronomicznego — KI-412
- Uniwersalny moduł ze standardowym blokiem współpracy z magistralą — KI-424
- Przetwornik częstotliwość-cyfra — KI-470
- Układ kalkujący (8-kanałowy, 15-bitowy przetwornik czas-cyfra) — KI-476
- Multipleksjer — moduł rozszerzenia kanału we-wy — KI-500A
- Sterownik kasetowy rozszerzonego kanału — KI-500B

### Moduły wprowadzania i wyprowadzania sygnałów cyfrowych

- Uniwersalny dwukierunkowy rejestr 8-bitowy — KI-420
- 32 wejścia przerywające dwustanowe w wersjach z separacją galwaniczną i bez — KI-432



Rys. 1. Struktura sprzętowa rejestratora PRS-4



# Spis treści rocznika 1982

nr str.		nr str.	
<b>ARTYKUŁY PROBLEMOWE</b>			
<b>BISKUPSKI PAWEŁ</b> — Analiza sygnatur — uniwersalna metoda wykrywania uszkodzeń w układach cyfrowych		8-9	11
<b>BULANDRA MAGDALENA, KWIATKOWSKI MAREK, MAJ WALDEMAR</b> — Łączenie maszyn cyfrowych Odra/ICL, 1904		7	9
<b>CHWASZCZEWSKI STEFAN i in.</b> — p. System CAMAC			
<b>CZERMAK ADAM, JABŁOŃSKI JÓZEF</b> — p. System CAMAC			
<b>DAWIDOWSKI JAN, OW CZARCZAK PIOTR</b> — Zastosowanie pakietu PSL/PSA do prowadzenia słownika-skorowidza bazy danych		2-3	21
<b>DAWIDOWSKI JAN, OW CZARCZAK PIOTR, WISNIEWSKI MAREK</b> — Dwie metody badania sprawności systemów komputerowych		8-9	7
<b>DRASZANOWSKI HENRYK, SAWICKI KRZYSZTOF</b> — p. System CAMAC			
<b>DZIEGLEWSKI GRZEGORZ</b> — p. System CAMAC			
<b>FLADROWSKA EMILIA</b> — Czas systemów przetwarzalnych		2-3	15
<b>FRYŃ RYSZARD</b> — Doświadczenia i problemy programowania aplikacyjnego w FSO		1	7
<b>GERWIN KRZYSZTOF, SUKIENNIK JERZY</b> — Zastosowanie systemu operacyjnego R 800 na standardowym zestawie MERA 9150		6	20
<b>HUZAR ANDRZEJ</b> — Weryfikacja programów — podstawowe pojęcia		2-3	13
<b>KAPUŚCIK WALDEMAR</b> — Eksploatacja baz danych w hucie „Szopienice”		1	11
<b>KASMAN LEON</b> — Współdziałanie resortowego systemu informatycznego MHiPM z innymi krajowymi systemami handlu zagranicznego		8-9	22
<b>KAZUBEK MARIAN i in.</b> — p. System CAMAC			
<b>KOBA TERESA, BORSUK STANISŁAW</b> — p. System CAMAC			
<b>KUKUŁA ANDRZEJ, SZMIDT URSZULA</b> — Analiza przydatności USZBD RODAN dla baz danych o wysokiej dyspozycyjności		8-9	16
<b>KRECZMAR ANTONI, SALWICKI ANDRZEJ</b> — Język programowania LOGLAN		7	4
Część 1		8-9	4
Część 2			
<b>MIRKOWSKI JACEK i in.</b> — p. System CAMAC			
<b>MOKROSZ MAREK, SUCHY JANUSZ, ZYMEŁKA KRYSZTOF</b> — Programowany rejestrator PRS-4 — podstawa automatyzacji kopalni węgla kamiennego		8-9	19
<b>NOSOWSKI WIESŁAW, ZAJĄCZKOWSKI PIOTR</b> — Zajęcia z podstaw informatyki		6	4
<b>RAKOWSKI LESZEK</b> — AESN — pakiet programów do generowania wersji nakładanych programów fortranowych dla ODRY 1305		8-9	27
<b>RYZMAR ZYGMUNT</b> — S&DL — język specyfikacyjny do projektowania strukturalnego (zarys propozycji)		2-3	18
<b>SMERECZYŃSKI ANDRZEJ</b> — Automatyzacja prac wydawniczych na IBM/370		7	18
<b>SNOWACKA MIROSLAWA, ZIEBIŃSKI MICHAŁ</b> — Automatyczne przetwarzanie tekstów		7	15
<b>SOBCZYK MAREK</b> — Algorytmy sortowania bez porównań		8-9	14
<b>SONDEJ HALINA, WISNIEWSKI JANUSZ</b> — MOLATO — System wspomagający nauczanie		2-3	24
<b>STARZYK HANNA</b> — OSIRIS III w badaniach statystycznych		6	14
<b>STOKALSKI ANDRZEJ</b> — Projektowanie i budowa systemów informatycznych. Organizacja cyklu rozwojowego		2-3	8
<b>ŚWIAĆ STEFAN, WEINER RYSZARD</b> — DISCAR — preprocesor języka FORTRAN na maszynie cyfrową CYBER		7	23
<b>WALCZAK TADEUSZ</b> — Informatyka w okresie przemian		1	4
<b>WIECZORKOWSKI KAZIMIERZ</b> — REKRUTACJA — system kwalifikowania kandydatów na studia		6	17
<b>WINIARSKI MACIEJ S. (oprac.)</b> — Styl programowania w języku FORTRAN		6	12
<b>ZALEWSKI JANUSZ</b> — ADA — nowy język programowania			
2) Jednostki programowe i instrukcje		1	15
3) Typy i inne konstrukcje językowe		2-3	4
4) Przykład programowania systemu CAMAC		4-5	17
5) Krytyka języka		6	10
<b>ZAWIŁA-NIEDŹWIECKI JANUSZ</b> — CHANGE — uniwersalny pakiet korygowania zbiorów		7	20
<b>ZĘBAŁA ANDRZEJ</b> — Weryfikacja struktury logicznej bazy danych		7	12
<b>ZĘBROWSKI JACEK</b> — Komputery osobiste		6	6
<b>SYSTEM CAMAC</b>			
<b>CHWASZCZEWSKI STEFAN, BŁASZCZĘC MIROSLAW, JABŁOŃSKI KONRAD, JEZIERSKA HANNA, KILIŃ STANISŁAW, KOŚLACZ STANISŁAW, PINDARA ANNA, RUSINOWSKI ZDZISŁAW, SZMEK WIESŁAW</b> — Komputerowy system wspomagania dyspozytora bloku KSWDB-360		4-5	10



	nr	str.		nr	str.
<b>CZERMAK ADAM, JABŁOŃSKI JÓZEF</b> — Wielofunkcyjny system spektrometryczny CA-DOS	4-5	6	Dlaczego nie korzystamy z oprogramowania powtarzalnego — Grażyna Klajn-Zienkiewicz	7	27
<b>DRASZANOWSKI HENRYK, SAWICKI KRZYSZTOF</b> — Oprogramowanie diagnostyczne bloków zestawu CAMAC	4-5	23	54 Międzynarodowe Targi Poznańskie. Inwazja mikroprocesorów — Jacek Zebrowski	7	28
<b>DZIEGLEWSKI GRZEGORZ</b> — Zestaw CAMAC do przemysłowych badań izotopowych	4-5	20	Zastosowania komputerów w przemyśle — Janusz Zalewski	8-9	32
<b>KAZUBEK MARIAN, JAMROGIEWICZ TOMASZ, ROTTER SŁAWOMIR</b> — Sprzężenie magistrali CAMAC z magistralą GPIB	4-5	15	Kadra kierownicza PGR o potrzebie zastosowań informatyki — Jadwiga Orylska	8-9	34
<b>KOBA TERESA, BORSUK STANISŁAW</b> — Sterowniki równoległych, wielokasetowych i wieloprocessorowych zestawów CAMAC	4-5	13	Narzędzie ekonomisty? (J. G.)	8-9	35
<b>MIRKOWSKI JACEK, PIĄTKOWSKI ADAM, PIĄTKOWSKA EWA</b> — Zestaw ELZA do określania parametrów dynamicznych procesów technologicznych	4-5	22	<b>ZJEDNOCZENIE INFORMATYKI</b>		
<b>ZALEWSKI JANUSZ</b> — ADA — nowy język programowania (4). Przykład programowania systemu CAMAC	4-5	17	Badania ankietowe użytkowników systemów informatycznych — Kazimierz Dudek, Józef Dziejicki	2-3	31
			Zjednoczenie Informatyki — podsumowanie działalności, kierunki zmian — Tomasz Pawlak	4-5	33
			Systemy automatycznego rozliczania użytkowników — Wiesław Sojka	4-5	37
<b>ALGORYTMY</b>			<b>POLSKIE TOWARZYSTWO INFORMATYCZNE</b>		
Procedura wyszukująca dany wzorzec w tekście — Andrzej Szalas, Zbigniew Świrski	1	18	Bieżąca działalność PTI — Bolesław Szymański	1	25
Procedura generująca drzewa etykietowane — Andrzej Szalas, Zbigniew Świrski	2-3	26	Z prac PTI (B. O.)	2-3	30
Dwa algorytmy sortowania wewnętrznego — Andrzej Szalas, Zbigniew Świrski	4-5	25	<b>ZE ŚWIATA</b>		
Dwa sposoby obsługi tablic rozproszonych — Andrzej Szalas, Zbigniew Świrski	6	23	Diagnostyka raz jeszcze — Andrzej Hławiczka	1	26
O problemie wyboru — Andrzej Szalas, Zbigniew Świrski	7	25	Komputery w nauczaniu — Mieszysław Baze-wicz	1	28
Sortowanie przez podział dystrybucyjny — Marek Sobczyk, Andrzej Szalas, Zbigniew Świrski	8-9	29	Bardzo Wielkie Bazy Danych'81 — Jan Chomici	1	28
O pewnym rozwiązaniu układu równań liniowych — Danuta Kalinowska	8-9	31	Przemysł komputerowy USA w roku 1980 — oprac. Jan Ryżko	2-3	34
			Dysk optyczno-numeryczny zamiast mikrofilmowania? — oprac. P.A. Milewski	2-3	36
<b>Z KRAJU</b>			CAMAC na pokładzie promu COLUMBIA? — Janusz Zalewski	4-5	41
Informatyka na studiach pielęgniarskich — Zbigniew Kolat	1	19	Nie tylko CAMAC — Janusz Zalewski	4-5	42
Zastosowanie informatyki w projektowaniu budownictwa — Maciej Robakiewicz	1	20	Systemy rozproszone — Monachium'81 — Janina Mincer	4-5	44
Oczami studenta — Joanna Gutt	1	22	Europejskie Stowarzyszenie Usług Informatycznych (E. K.)	4-5	47
Co dalej ze związkami zawodowymi — Bogdan Fiutowski	1	24	Prawda, która może urazić — Edsger W. Dijkstra	6	26
Sytuacja kadry w ośrodkach informatyki — Grażyna Klajn-Zienkiewicz	2-3	28	Kongres IFIP'83 — oprac. Władysław Klepacz	6	28
Komitet ds. Systemu CAMAC (J. Z.)	4-5	27	Macierzowy model gospodarki — Adam B. Empacher	6	31
KHBBT — 20 lat współpracy naukowej — Juliusz Lech Kulikowski	4-5	28	SICOB'81 — Danuta Segiet	7	31
Dyrektor z konkursu (J. G.)	4-5	29	Japonia — Piąta generacja — oprac. Marianna Sobczyk	7	32
Katalog produktów programowych i systemów informatycznych — Leszek Sankowski	4-5	31	DATASHOW'82 — oprac. Marianna Sobczyk	8-9	36
			Informatyka we Francji — kilka liczb — oprac. Danuta Segiet	8-9	36



## TERMINOLOGIA

	nr	str.
Terminologia języka ADA — Janusz Zalewski	1	30
Terminologia języka A A (c.d.) — Janusz Zalewski	2-3	39
Systematyzacja pojęć i terminów używanych i obowiązujących w systemie CAMAC — Krzysztof Rzymkowski	4-5	48
Terminologia języka ADA (dokończenie) — Janusz Zalewski	6	32
O programowaniu podstawowym — Janusz Zalewski	7	34
O testowaniu, weryfikacji i uwiarygodnieniu programów — Janusz Zalewski	8-9	38

## RECENZJE

Sterowanie komunikacją w sieciach komputerowych — Mieczysław Bazewicz	2-3	37
Obieg informacji gospodarczej — Andrzej Sokolowski	4-5	47
Kombinatoryka dla programistów — Andrzej Szalas	7	36

## LISTY

Kilka uwag o „ALGORYTMACH” — Jacek Zebrowski	2-3	40
Trudne chwile informatyki? — Bartłomiej Kruszelnicki	2-3	III okł.
O języku służącym do formułowania specyfikacji — Jan Dawidowski	6	32
Jak rozruszać informatykę? — Ryszard Grzesiak	7	34

## POGLĄDY

Całkowita decentralizacja — Czesław Syc	1	32
Informatyka w przedsiębiorstwie przemysłowym — Czesław Rybak	4-5	52
Selekcja negatywna — Janusz Gwiazda	8-9	III okł.
Ot, marzenie — Zbigniew Gluza	8-9	III okł.

## KONFERENCJE

FTS&D'82	1	27
Convention Informatique'82	2-3	30
Międzynarodowe seminarium CAMAC'83	4-5	12
INTERKOMPUTO'82	4-5	46
COMPCONTROL'83	4-5	50
Wspomagane projektowanie systemów	6	27
Konferencje specjalistyczne IFIP	7	8
CAPE'83	7	14
SPIS'82 (A. Sokolowski)	8-9	18
MEDINFO'83	8-9	26

Mokrosz M., Suchy J., Żymelka K.: Programowany rejestrator PRS 4 — podstawa automatyzacji kopalni węgla kamiennego  
INFORMATYKA 1982, nr 8-9, s. 19

Charakterystyka sprzętowa i programowa rejestratora skonstruowanego przez zaplecze naukowo-badawcze górnictwa. Podano szeroki zakres zastosowań w kopalniach węgla kamiennego wskazując, że rozwiązania konstrukcyjne i własności eksploatacyjne rejestratora uzasadniają jego użycie również w innych gałęziach gospodarki i dziedzinach zastosowań.

Kasman L.: Współdziałanie resortowego systemu informatycznego MHiPM z innymi krajowymi systemami handlu zagranicznego  
INFORMATYKA 1982, nr 8-9, s. 22

Charakterystyka resortowego systemu informatycznego z dziedziny handlu zagranicznego z podkreśleniem jego powiązań z innymi systemami handlu zagranicznego opartych na wymianie informacji na taśmach magnetycznych w warunkach bardzo zróżnicowanego sprzętu komputerowego.

Rakowski L.: AESN — pakiet programów do generowania wersji nakładanych programów fortranowych dla ODRY 1305  
INFORMATYKA 1982, nr 8-9, s. 27

Charakterystyka pakietu zrealizowanego na komputery Odra 1305 z systemem operacyjnym GEORGE 3 w Biurze Projektów PROSYNCEM w Gliwicach. Omówiono potrzeby opracowania, zadania i możliwości, budowę, metodę realizacji algorytmu oraz zastosowania tego pakietu.

Мокрош М., Сухы И., Жымелка К.: Програмный регистратор PRS 4 — база автоматизации каменноугольной шахты  
INFORMATYKA 1982, № 8-9 стр. 19

Характеристика программная и оборудовательная регистратора конструированного научно-исследовательской базой горной промышленности. Представлен широкий диапазон употреблений в каменноугольных шахтах, указывая, что конструкционные решения и эксплуатационные качества регистратора, обосновывают его применение также в других отраслях промышленности и областях применений.

Касман Л.: Взаимодействие ведомственной системы вычислительной техники Министерства металлургии и машиностроительной промышленности с другими отечественными системами внешней торговли  
INFORMATYKA 1982, № 8-9 стр. 22

Характеристика ведомственной системы вычислительной техники внешней торговли, с указанием на её связь с другими системами внешней торговли, исходя из обмена информацией на магнитных лентах, в условиях очень неоднородного хардвера.

Ракowski Л.: AESN — пакет программ для генерирования наложений фортрановских программ для ODRY 1305  
INFORMATYKA 1982, № 8-9 стр. 27

Характеристика пакета реализованного для вычислительной машины Odra 1305 с операционной системой GEORGE 3 в Проектбюро PROSYNCEM в Gliwicach. Обсуждено необходимость разработки, задачи и возможности, конструкцию, метод реализации алгоритма и применение этого пакета.



Mokrosz M., Suchy J., Żymelka K.: Programable logger PRS 4 — a base for coal mines automation  
INFORMATYKA 1982, No. 8-9, p. 19

Hardware and software characteristics of the logger built by the research institutions of polish coal mining industry. Wide scale of applications in coal mines is presented emphasising, that the constructional solutions and operating features of the logger motivate its use also in other national economy branches and application areas.

Kasman L.: Cooperation of the metallurgy and machinery industry departmental EDP system with other national foreign trade systems  
INFORMATYKA 1982, No. 8-9, p. 22

Characteristics of the departmental EDP system in the field of foreign trade, emphasising its interfaces with other national foreign trade systems, supported on magnetic tape exchange in conditions of very differentiated hardware.

Rakowski L.: AESN — a program package for generating overlapping versions of FORTRAN programs on ODRA 1305 computers  
INFORMATYKA 1982, No. 8-9, p. 27

Characteristics of the package for ODRA 1305 computers with GEORGE 3 operating system, realized in the designing bureau PROSYNCHM in Gliwice. Elaboration need, tasks and possibilities, structure, algorithm realization method and applications of the package are discussed.

Mokrosz M., Suchy J., Żymelka K.: Programmierbarer Registrierer PRS 4 — eine Grundlage der Automatisierung von Steinkohlengruben  
INFORMATYKA 1982, Nr. 8-9, S. 19

Eine Charakteristik von Hard- und Software eines Registrierers, der von Forschungseinheiten des polnischen Bergbaus konstruiert wurde. Eine breite Palette von Anwendungen in Steinkohlengruben zeigt an, dass die Konstruktionslösungen und die Betriebseigenschaften des Registrierers seine Nutzung auch in anderen Wirtschaftszweigen und Anwendungsbereichen begründen.

Kasman L.: Zusammenwirken des ministeriellen EDV-Systems von Hütten- und Maschinenindustrie mit anderen einheimischen Systemen des Aussenhandels  
INFORMATYKA 1982, Nr. 8-9, S. 22

Eine Charakteristik des ministeriellen EDV-Systems aus dem Bereiche des Aussenhandels mit Betonen seiner Verknüpfungen mit anderen Systemen des Aussenhandels. Diese Verknüpfungen wurden auf Grund des Informationsaustausches mittels Magnetband und unter Bedingungen einer sehr stark differenzierten Hardware realisiert.

Rakowski L.: AESN — ein Programmpaket zur Generierung der Überlagerungsversionen von FORTRAN-Programmen für ODRA 1305  
INFORMATYKA 1982, Nr. 8-9, S. 27

Eine Charakteristik des Pakets, das für ODRA 1305 Rechner mit Betriebssystem GEORGE 3 in dem Projektierungsbüro PROSYNCHM in Gliwice erarbeitet wurde. Es wurden Erarbeitungsbedürfnis, Aufgaben und Möglichkeiten, Bauweise, Methode der Algorithmusrealisation und Anwendungen des Pakets besprochen.

## Nowe zasady prenumeraty

Zamówienia i przedpłaty na prenumeratę obok wymienionych czasopism przyjmuje bezpośrednio Wydawnictwo Czasopism i Książek Technicznych SIGMA:

ADRES pocztowy: Wydawnictwo SIGMA, skrytka 1004, 00-950 Warszawa  
KONTO bankowe: nr 1036-7490-139-11 III O/M NBP Warszawa

**Jednostki gospodarki uspołecznionej, instytucje i organizacje** przesyłają zamówienia (w 1 egz.) zawierające: tytuł (tytuły) czasopisma, liczbę zamawianych egz. poszczególnych tytułów, okres prenumeraty oraz pełny adres zamawiającego wraz z kodem pocztowym, ewent. adresy odbiorców, którzy na zlecenie zamawiającego mają otrzymywać przesyłki, a także numer konta bankowego zamawiającego.

!!! Dopisując w zamówieniu PRENUMERATA STAŁA, zamawiający nie będzie musiał corocznie ponawiać zamówienia, a jedynie dokonywać przedpłaty wg aktualnie obowiązujących cen na wezwanie Wydawnictwa !!!

Warunkiem realizacji zamówienia jest równoczesne dokonanie odpowiedniej przedpłaty na ww. konto Wydawnictwa SIGMA.

**Prenumeratorzy indywidualni** dokonują przedpłaty przekazem na ww. konto podając na odwrocie odcinka dla adresata- posiadacza rachunku: tytuł czasopisma, liczbę zamawianych egzemplarzy oraz okres prenumeraty.

**Przedpłaty przyjmowane są w terminach:**

- DO 5 GRUDNIA 1982 r. — na I kwartał, I półrocze i cały rok 1983 oraz na prenumeratę stałą (wieloletnią),
- do 10 marca — na II kwartał,
- do 10 czerwca — na III kwartał i II półrocze,
- do 10 września — na IV kwartał,
- do 25 listopada — na I kwartał, I półrocze i cały rok następny oraz na prenumeratę stałą (wieloletnią).

**UWAGA:** Obowiązuje bardzo czytelne pismo i podawanie kodu pocztowego.



● 51 wejścia dwustanowe nieprzerwywane w dwóch wersjach: z separacją galwaniczną i bez — KI-450

● 61 wyjścia dwustanowe w dwóch wersjach: z separacją galwaniczną i bez — KI-460

### Moduły wprowadzania i wyprowadzania sygnałów analogowych

● 8-kanalowy specjalistyczny przetwornik napięcie-czas — KI-473; zakres napięciowy:  $-5 - +5$  V, wyjście dwu-stanowe w konwencji TTL wyznaczające interwały czasowe proporcjonalne do bezwzględnej wartości wielkości mierzonej w każdym kanale:  $0 \leq T_i \leq 25,6 \mu s$

● 8-kanalowy przetwornik A/C — dwa moduły KI-473 i KI-477; zakres napięciowy:  $-5 - +5$  V, dokładność: 8-bitów z bitem przepelnienia i z bitem znaku, czas przetwarzania:  $51,2 \mu s$ , zmienna częstotliwość próbkowania zadawana programowo, max. częstotliwość: 10 kHz

● 8-kanalowy specjalistyczny przetwornik napięcie/czas — dwa moduły KI-473 i KI-474; zakres napięciowy:  $-5 - +5$  V, wyjścia dwustanowe w konwencji TTL wyznaczające interwały czasowe proporcjonalne do bezwzględnej wartości sygnału wejściowego w każdym kanale:  $0 \leq T_i \leq 25,6 \mu s$ ; sygnalizuje przekroczenia progu symetrycznego względem zera, zadawanego programowo ze strefy histerezy — definiując adres kanału i znaku wartości chwilowej ostatniego i poprzedniego przejścia przez strefę histerezy; maksymalna częstotliwość próbkowania: 10 kHz

● 8-kanalowy specjalistyczny przetwornik A/C z pamięcią — cztery moduły  $2 \times$  KI-475, KI-473; zakres napięciowy:  $-5 - +5$  V, dokładność: 8 bitów, pamiętanie maksymalnych wartości przeregulowań względem zera, możliwość pobierania chwilowych wartości sygnału, sygnalizacja przekroczenia progu symetrycznego względem zera, programowo zadawanego ze strefą histerezy — definiując adres kanału i znaki wartości chwilowej ostatniego i poprzedniego przejścia przez strefę histerezy, maksymalna częstotliwość próbkowania: 10 kHz

● kompensacyjny przetwornik A/C — KI-480; zakres napięcia:  $0 - +5$  V, dokładność: 8 bitów oraz bit przepelnienia, czas przetwarzania:  $25,6 \mu s$ , cztery komutowane wejścia

● integracyjny wolnozmienny przetwornik A/C — KI-482; zakres napięciowy:  $-10 - +10$  V, dokładność: 12 bitów oraz bit znaku, podwójne całkowanie, pełna synchronizacja z częstotliwością sieci, czas przetwarzania:  $\leq 80 \mu s$ , wejście niesymetryczne

● wzmacniacz pomiarowy z programowym wzmocnieniem do modułu KI-482 — KI-483; zakres napięciowy:  $-10 - +10$  V, programowo zadawane wzmocnienie: 1, 2, 5, 10, 20, 50, 100, 200, 500, 1000, wejście symetryczne, wyjście niesymetryczne separowane galwanicznie od obwodów wejściowych wzmacniacza

● 16-kanalowy skaner stykowy z 16 na 1 — KI-484; zakres napięć wejściowych:  $-10 - +10$  V, max. prąd: 125 mA

● 8-kanalowy skaner z pamięcią analogową i detekcją progu w każdym kanale — KI-486; zakres napięcia:  $0 - +5$  V, wejścia niesymetryczne

● 8-kanalowy multiplekser z pamięcią, programową zmianą progu dla każdego kanału — KI-487; zakres napięcia:  $0 - +5$  V, wartości progów: 0,5; 1,5 i 2 V, wejścia niesymetryczne.

Wymienione moduły umożliwiają kompletowanie konfiguracji rejestratora PRS-4 zgodnie z wymaganiami użytkownika. W zakresie przetwarzania sygnałów analogowych użytkownik ma możliwość zestawienia tradycyjnych układów pomiarowych złożonych ze skanerów i przetworników A/C (np. moduły KI-486 — z modułem KI-480 lub moduły KI-484 — z modułami KI-482 i KI-483), a także specjalistycznych urządzeń pomiarowych wielomodułowych (np. specjalistyczny układ pomiaru sygnałów sejsmicznych, złożony z modułów KI-473, KI-474,  $2 \times$  KI-475 i KI-477).

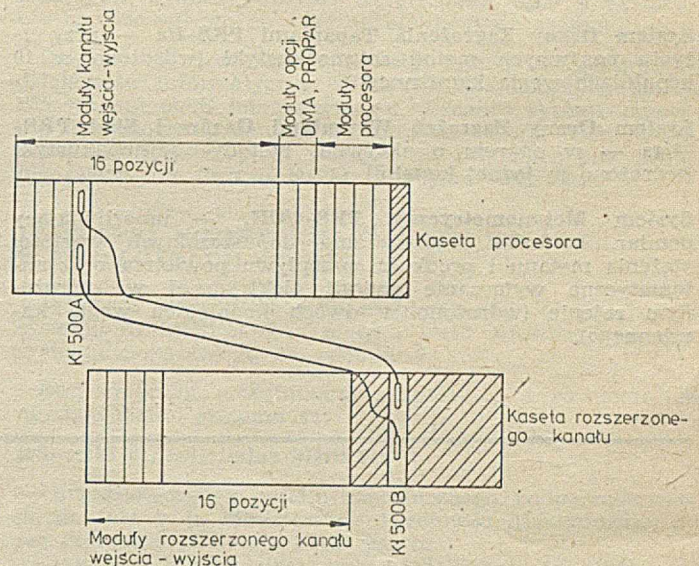
Konfigurowanie kanału we-wy rejestratora PRS-4 może się odbywać w dwóch poziomach:

● pierwszy — polega na wykorzystaniu opracowanych i produkowanych modułów

● drugi — polega na samodzielnym wykonaniu modułów w oparciu o uniwersalny moduł KI-424, zawierający standardowy blok współpracy z magistralą I/O; użytkownik projektuje tylko specjalistyczną część modułu związaną z rozwiązywaniem problemem, nie zajmując się rozwiązaniem współpracy z procesorem; poziom ten zasługuje na szczególne podkreślenie, daje bowiem każdemu użytkownikowi możliwość samodzielnego wykonania spe-

cialnych modułów kanału we-wy — umożliwi to w wielu wypadkach uproszczenie układów sprzęgających rejestrator z obiektem.

W przypadku obiektu wymagającego rozbudowanego kanału we-wy możliwe jest wykorzystanie modułu multipleksera KI-500A. Umożliwia on wraz z modułem sterownika kasety KI-500B rozszerzenie kanału we-wy o dalsze 16 pozycji. Sposób rozszerzenia kanału we-wy pokazano na rysunku 2.



Rys. 2. Rozmieszczenie modułów rejestratora PRS-4 w kasce oraz sposób rozszerzenia kanału wejścia-wyjścia

### STANDARD WYKONANIA REJESTRATORA

Wszystkie moduły elektroniczne rejestratora wykonane są w mechanicznym standardzie CAMAC ze złączami pośrednimi 831/841, jako złączami okresowania stałego, oraz złączami szufladowymi, jako złączami okablowania czołowego. Moduły umieszcza się w 19-calowych kasetach typu CAMAC.

Rozmieszczenie pakietów przedstawiono na rysunku 2. Kasety umieszczone są w stojakach urządzeń cyfrowych. Do konstrukcji rejestratora użyto wyłącznie krajowych elementów TTL podstawowej i średniej skali integracji, montowanych na obwodach dwustronnie drukowanych. Rejestrator PRS-4 produkowany jest przez Zakład Elektroniki Górniczej w Tychach wchodzący w skład Centrum Naukowo-Produkcyjnego Elektroniki i Automatyki Górniczej EMAG z siedzibą w Katowicach.

### OPROGRAMOWANIE

Rejestrator PRS-4 wyposażony jest w bogate oprogramowanie podstawowe i systemowe. Daje to użytkownikowi szerokie możliwości projektowania i wykonania oprogramowania użytkowego związanego z własnymi wymaganiami.

Oprogramowanie podstawowe zawiera:

- assembler (minimalna wymagana pojemność pamięci — 4 K słów)
- języki wyższego rzędu: FORTRAN, BASIC, ALGOL (minimalna wymagana pojemność pamięci — 8 K słów)
- programy uruchamiania (debugery)
- programy edycyjne
- programy testujące
- biblioteka podprogramów standardowych.

Oprogramowanie systemowe zawiera:

- system operacyjny czasu rzeczywistego, rezydujący w pamięci operacyjnej SOCR 79
- system dyskowy BABEL do przygotowywania programów, oparty o pamięć na dyskach elastycznych
- system operacyjny BCS — jednozadaniowy.



## PRZYKŁADY ZASTOSOWAŃ

Rejestrator PRS-4 stanowi jednostkę centralną projektowanych w CNP EMAG systemów nadzoru dyspozytorskiego. Na jego bazie opracowano i wdrożono do produkcji następujące systemy:

**System Kontroli Parametrów Produkcji PRS-4/p** — zapewniający kontrolę pracy urządzeń wydobywczo-transportowych i bilansowanie wydobycia (wdrożono w sześciu kopalniach węgla kamiennego)

**System Oceny Zagrożenia Tąpniętami PRS-4/a** — przy użyciu pasywnych metod sejsmoakustyki (wdrożono w 10 kopalniach węgla kamiennego)

**System Oceny Zagrożeń Wyrzutami Gazów i Skał PRS-4/gs** — w oparciu o aktywne metody sejsmoakustyki (wdrożono w jednej kopalni)

**System Metanometryczny PRS-4/CH<sub>4</sub>** — umożliwiający pomiar, kontrolę przekraczania dopuszczalnych wartości stężenia metanu i prędkości przepływu powietrza oraz automatyczne wyłączanie energii elektrycznej w zagrożonym rejonie (wdrożono w dwóch kopalniach węgla kamiennego).

Ponadto w trakcie badań lub opracowania znajdują się:

**System Lokalizacji Wstrząsów PRS-4/m** — umożliwiający określenie miejsca wystąpienia wstrząsu i jego energii przy użyciu metody mikrosejsmologii

**System Wczesnego Wykrywania Pożarów PRS-4/poż.** — umożliwiający wykrywanie pożarów endo- i egzogenicznych

**System Kontroli Ruchu Załogi PRS-4/z** — umożliwiający kontrolę i rejestrację ruchu załogi.

\* \* \*

Wieloletnia eksploatacja wdrożonych systemów wykazała, że rejestrator PRS-4 charakteryzuje się wysoką niezawodnością w trudnych warunkach kopalni węgla kamiennego (systemy bezpieczeństwa), prostotą konfiguracji, bogatym asortymentem modułów kanału przemysłowego — cyfrowych i analogowych.

Cechy te predysponują rejestrator PRS-4 do zastosowań również w innych dziedzinach gospodarki — do kontroli i sterowania produkcją, ograniczonego zakresu przetwarzania danych, wspomaganie prac eksperymentalnych w laboratoriach oraz obliczeń inżyniersko-technicznych.

LEON KASMAN  
Instytut Organizacji Przemysłu Maszynowego ORGMASZ  
Warszawa

# Współdziałanie resortowego systemu informatycznego MHiPM z innymi krajowymi systemami handlu zagranicznego

Dotychczasowy rozwój krajowych zastosowań informatyki dla celów zarządzania charakteryzuje się m.in. mnogością systemów eksploatowanych w obszarze tej samej dziedziny. I choć postulowana samodzielność jednostek gospodarczych w oczywisty sposób może pobudzać je do stosowania rozwiązań najbardziej odpowiadających ich własnym potrzebom, to jednak przyjęcie w pewnym zakresie rozwiązań spójnych i wzajemnie komplementarnych może wydatnie zwiększyć efektywność zastosowanych przez nie systemów.

Niniejszy artykuł ilustruje powyższą tezę na przykładzie systemu informatycznego handlu zagranicznego (hz), realizowanego nieprzerwanie od 1974 r. w jednostkach organizacyjnych resortu przemysłu maszynowego, a po reorganizacji resortów w 1981 r. — rozszerzonego na potrzeby całego resortu hutnictwa i przemysłu maszynowego (MHiPM). Celem systemu jest usprawnienie procesów podejmowania decyzji, przez dostarczanie poszczególnym ogniwom zarządzania niezbędnych informacji dotyczących hz.

System daje kompleksowy obraz sytuacji w handlu zagranicznym z wyodrębnieniem eksportu oraz importu, zarówno dla całego resortu, jak i podległych mu jednostek.

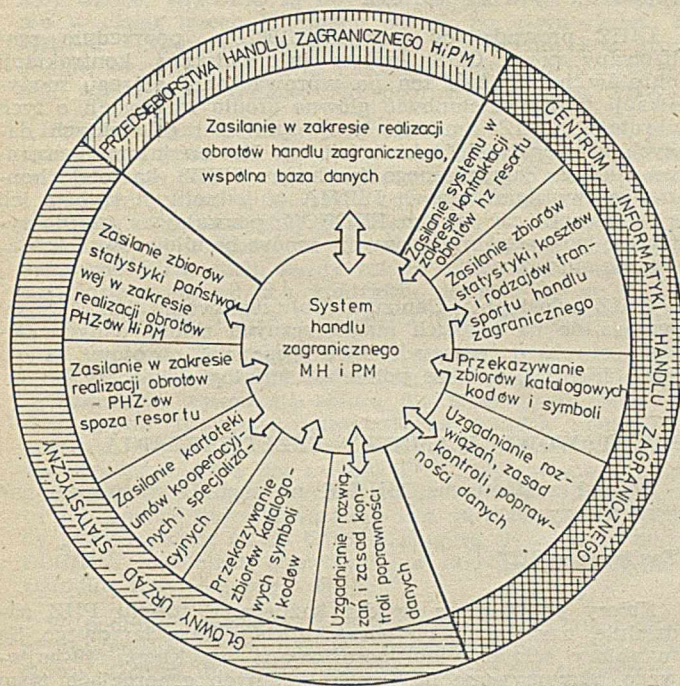
Korzysta z danych o operacjach dokonywanych na rzecz jednostek resortu HiPM w sześciu przedsiębiorstwach handlu zagranicznego (PHZ) działających do 1981 r. w b. Ministerstwie Przemysłu Maszynowego, a także w pozostałych PHZ w zakresie obsługi MHiPM.

Z myślą o stworzeniu pełnej bazy danych MHiPM w Instytucie Organizacji Przemysłu Maszynowego (IOPM), zaprojektowano i zrealizowano bezpośrednie współdziałanie systemu informatycznego MHiPM z innymi krajowymi systemami hz:

- państwowym systemem informatycznym statystyki handlu zagranicznego (SISHZ), realizowanym przez GUS
- resortowym systemem informatycznym handlu zagranicznego, realizowanym przez Centrum Informatyki Handlu Zagranicznego (CIHZ)
- obiektowymi systemami informatycznymi eksploatowanymi w sześciu PHZ należących do MHiPM. Współdziałanie to polega na:
  - okresowej (dekadowej lub miesięcznej) wymianie danych na taśmie magnetycznej (TM) między systemem hz MHiPM a wymienionymi wyżej krajowymi systemami hz
  - zdalnym, sukcesywnym przekazywaniu danych z sześciu PHZ — należących do MHiPM — z terminali bądź systemów minikomputerowych do komputera w Centrum Obliczeniowym (CO) HiPM.



Zakres tematyczny współdziałania systemu MHiPM z systemami otoczenia pokazany jest na rysunku 1.



Rys. 1. Schemat poglądowy współdziałania systemu informatycznego handlu zagranicznego MHiPM z systemami otoczenia

Przy projektowaniu systemu przyjęto założenie, że dane o obrotach winny pochodzić wyłącznie z PHZ — a więc by system rejestrował operacje faktycznie zrealizowane, a nie np. wielkość zaplanowanej produkcji przeznaczanej na eksport. Uwzględnivszy fakt istnienia sprawnego systemu informatycznego statystyki hz — realizowanego w skali kraju przez GUS i zasilanego ze wszystkich PHZ — przyjęto założenie, że system MHiPM zostanie wkomponowany do systemu GUS jako integralny jego fragment. Konsekwencją był podział zadań, polegający na przyjęciu przez system MHiPM wszystkich czynności związanych z rejestrowaniem, gromadzeniem, kontrolowaniem i dostarczaniem do Ośrodka Elektronicznego GUS na taśmie magnetycznej danych z realizacji obrotów sześciu podległych MHiPM PHZ. W zamian za to GUS w identyczny sposób zasila system danymi o obrotach resortu dokonywanych przez te PHZ, które nie podlegają MHiPM. Podobne rozwiązania przyjęto w zakresie współdziałania systemu

MHiPM z systemami realizowanymi na rzecz MHZ przez CIHZ.

Funkcjonowanie systemu MHiPM, jako integralnego fragmentu państwowego systemu hz, wprowadziło szereg ograniczeń w zawartości informacyjnej systemu, a także w harmonogramach przetwarzania. Należy podkreślić, że wzajemna wymiana zbiorów tworzonych w warunkach jednorazowej rejestracji danych w poszczególnych ośrodkach obliczeniowych PHZ (przy jednolitej bazie normatywnej katalogów, symboli, kursów walut, nazw itp.) w zasadniczy sposób wpływa na znaczną efektywność systemu.

Dzięki tym rozwiązaniom powstał system hz integrujący działalność informatyczną wielu instytucji dysponujących różnym sprzętem komputerowym. Ponadto system zapewnia dysponowanie danymi dotyczącymi całości obrotów zagranicznych resortu, bez względu na źródło ich powstawania (wewnątrz resortu HiPM oraz poza resortem).

### JEDNOSTKI UCZESTNICZĄCE I ICH ROLA W EKSPLOATACJI SYSTEMU

Jednostki uczestniczące w eksploatacji i korzystające z systemu hz MHiPM oraz obieg i treść informacji w tym systemie pokazane są na rysunku 2.

Bezpośrednimi uczestnikami eksploatacji i równocześnie użytkownikami systemu są:

#### jednostki organizacyjne MHiPM:

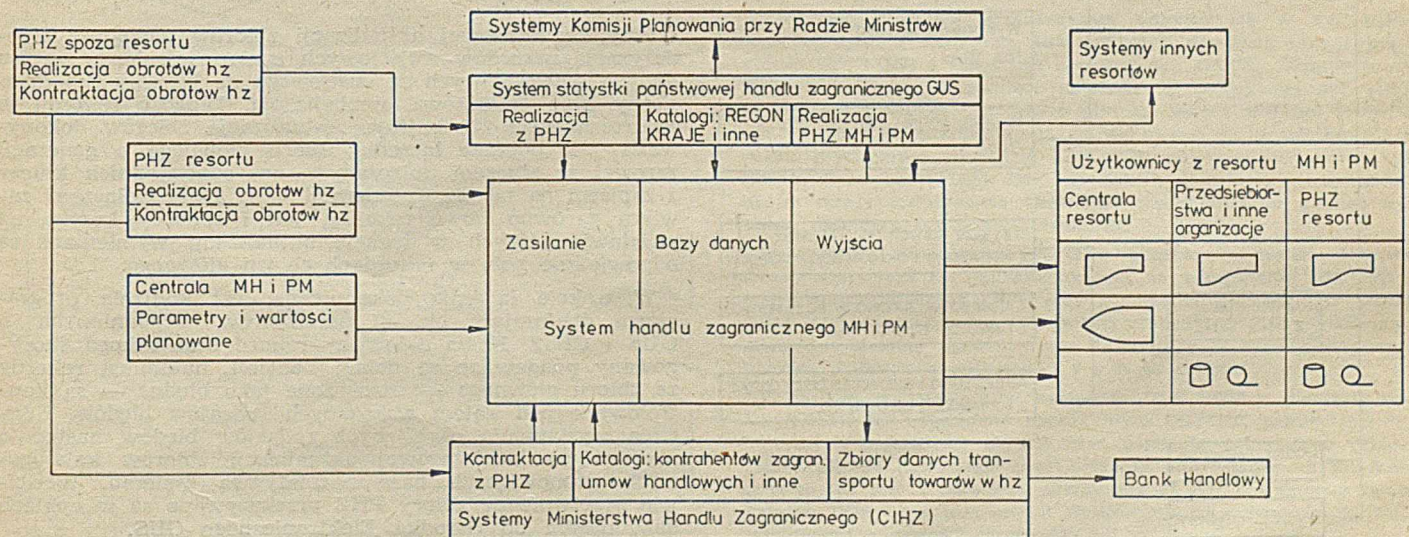
- przedsiębiorstwa handlu zagranicznego podporządkowane do 1981 r. b. Ministerstwu Przemysłu Maszynowego oraz pośrednio pozostałe PHZ resortu
- Centrala Ministerstwa, byle zjednoczenia (do 1981 r.), przedsiębiorstwa produkcyjne oraz niektóre biura studiów o charakterze ogólnoresortowym
- IOPM — będący realizatorem systemu z ramienia MHiPM (wykonawca projektu oprogramowania systemu oraz prac organizacyjno-wdrożeniowych, a następnie prowadzący eksploatację systemu w CO HiPM)

#### jednostki organizacyjne GUS:

- Departament Handlu Zagranicznego
- Ośrodek Elektroniczny GUS w Warszawie

#### Centrum Informatyki Handlu Zagranicznego.

PHZ są dla systemu głównymi, choć nie jedynymi, dostawcami danych o obrotach handlu zagranicznego. Dostarczają elementarnych danych opisujących w ramach jednolitego, standardowego rekordu każdą zrealizowaną dostawę eksportową i importową. Dane te, wraz z niezbędnymi korektami, zasilają zbiory własne każdego PHZ — zbiory założone i eksploatowane w CO HiPM. Zbiory te zaspokajają potrzeby własne PHZ, potrzeby statystyki



Rys. 2. Schemat funkcjonalny systemu handlu zagranicznego MHiPM i jego powiązań z systemami zewnętrznymi



państwowej (współdziałanie z systemem SISHZ GUS) oraz innych uczestników systemu. PHZ są w pełni odpowiedzialne za kompletność, wiarygodność i terminowość dostarczania danych do systemu. Każdy PHZ jest odpowiedzialny za stan i zawartość własnego zbioru.

Obecnie współdziała z systemem sześć PHZ branży elektromaszynowej: METAEXPORT, METRONEX, BUMAR, POLMOT, PEZETEL i UNITRA.

Ministerstwo HiPM oraz pozostałe jednostki organizacyjne resortu są głównymi odbiorcami informacji emitowanych przez system. System obsługuje przede wszystkim departamenty centrali MHiPM, zajmujące się bezpośrednio handlem zagranicznym (eksport, import i współpraca z zagranicą). Niemniej również inne departamenty (np. gospodarki materiałowej, techniki), a nawet przedsiębiorstwa produkcyjne wykorzystują informacje z baz danych systemu.

Centrala Ministerstwa jest głównym sponsorem rozwoju systemu; Centrala przyjmuje odpowiedzialność za właściwe sformułowanie i realizację porozumień regulujących współdziałanie systemu z jednostkami spoza resortu. We współpracy z MHZ, GUS opracowuje metodykę funkcjonowania systemu oraz całą bazę normatywną, w postaci nomenklatur, indeksów i katalogów, a także kontroli poprawności danych.

Dane o realizacji obrotów hz sześciu wymienionych wyżej PHZ są przekazywane z CO HiPM do Ośrodka Elektronicznego GUS — jako zbiory zarejestrowane na taśmie magnetycznej. W zamian za to, GUS udostępnia Centrum Obliczeniowemu HiPM kompletne zbiory danych o realizacji obrotów hz dokonywanych na rzecz MHiPM przez pozostałe przedsiębiorstwa hz. Zbiory te obejmują również dane o międzynarodowej kooperacji i specjalizacji produkcji, a także dotyczące programów operacyjnych, surowców i materiałów centralnie bilansowanych. GUS dostarcza przy tym na taśmach magnetycznych potrzebne zbiory nomenklatur, indeksów itp.

W ramach współpracy każdy z uczestników odpowiada za poprawność zbiorów w oparciu o uzgodnione parametry kontroli programowej, za właściwe ich zarejestrowanie na taśmie magnetycznej i za dostarczenie w uzgodnionych terminach. Podkreślić należy fakt, że w pracach nad

modyfikacjami i rozwojem SISHZ GUS zasięga opinii i rad uczestników systemu — bądź w formie spotkań roboczych, bądź tworząc zespoły problemowe z udziałem projektantów systemu MHiPM.

CIHZ prowadzi od początku 1982 r., poprzednio realizowany przez GUS, system informatyczny kontraktacji obrotów hz. System ten po wprowadzeniu szeregu modyfikacji powinien stanowić główne źródło informacji o tych obrotach. CIHZ realizuje przy tym szereg istotnych danych katalogowych, takich jak np. Nomenklatura Towarowa Handlu Zagranicznego (pochodna SWW), kartoteka kontrahentów zagranicznych FIRMA (z jednolitym trybem ich symbolizacji) czy system PROROK, pozwalający na identyfikację poszczególnych pozycji umów handlowych z krajami socjalistycznymi.

CIHZ udostępnia Centrum Obliczeniowemu HiPM zarejestrowane na taśmach magnetycznych zbiory danych elementarnych dotyczących kontraktacji i obrotów hz na rzecz MHiPM, a także potrzebne katalogi i indeksy.

## PODSTAWOWE ELEMENTY PRZETWARZANIA

Podział systemu hz MHiPM na moduły eksploatacyjne ilustruje rysunek 3.

### Zasilanie z PHZ

Zbiory prowadzone są dla każdego z sześciu PHZ oddzielnie, narastająco od początku roku kalendarzowego. Są to zbiory sekwencyjne, posortowane wg klucza 19-bajtowego, utrzymywane na trzech kolejnych generacjach taśm magnetycznych (0, —1, —2). W ten sposób uzyskano znaczną oszczędność zasobów pamięci dyskowej, a jednocześnie skuteczne zabezpieczenie przed ewentualnym zniszczeniem lub uszkodzeniem zbiorów.

W wyniku uzgodnień — dokonanych z PHZ, GUS i CIHZ — ujednolicono zawartość informacyjną oraz formę zapisu rekordów wejściowych (dla taśm magnetycznych, teletransmisji i kart dziurkowanych). Rekord dotyczy każdej dokonanej dostawy eksportowej i importowej oraz zawiera szczegółowy jej opis. Długość rekordu wejściowego wynosi 200 bajtów. Rekord obejmuje dane wartościowe, ilościowe i symbole opisujące, zgodnie z obowiązującą w hz jednolitą bazą normatywną:

dane wartościowe: wartość w cenach transakcyjnych, wartość w cenach sprzedaży, zakupu, marże, prowizje, koszty frachtu i ubezpieczenia.

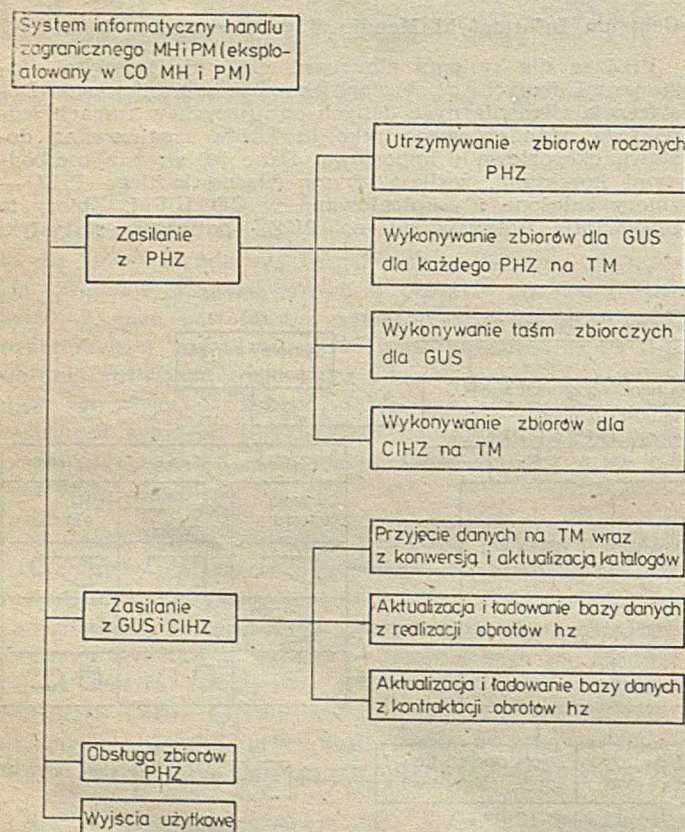
dane ilościowe: ilość w odpowiednich jednostkach miary oraz ciężary netto i brutto

dane opisowe: symbole towaru, jednostki miary, dane dostawcy i odbiorcy krajowego, dane kontrahenta zagranicznego, dane krajów sprzedaży lub zakupu, waluty, kategorii planu, pozycji umowy handlowej i kooperacyjnej, rodzajów frachtu, przejścia granicznego, wyróżnika ceny krajowej itp., a także symbol PHZ, biurka działu i sekcji realizującej dany kontrakt.

Przy każdorazowej aktualizacji zbiorów następuje standaryzacja rekordów wejściowych, zmiana struktury zapisu danych wartościowych ze znakowego na binarny lub dziesiętny oraz dodatkowa symbolizacja. Długość rekordu w zbiorze wynosi 220 bajtów. Aktualizacja zbiorów dokonywana jest poprzez łączenie zbioru głównego o generacji zerowej ze zbiorem dołączanym, np. odpowiednich kluczy z zapisem na taśmie o generacji — 2. Zbiór dołączany zawiera zarówno nowe rekordy, jak i rekordy korekt, na podstawie których w trakcie aktualizacji wymieniane są odpowiednie pola w rekordach zbioru głównego.

W trakcie łączenia dokonywana jest kontrola programowa obejmująca ok. 40 parametrów uzgodnionych z GUS i CIHZ. Nowo dołączany rekord oraz rekord skorygowany poddawane są pełnej kontroli, natomiast rekordy ze zbioru głównego — zaznaczone jako błędne — są kontrolowane pod kątem wskazanych rodzajów błędów. Kontrola wystąpienia określonych rodzajów błędów następuje również po każdorazowej aktualizacji zbiorów katalogowych, stanowiących bazę normatywną systemu. Właściwie przetworzone zbiory PHZ przekazywane są w cyklach dekadowych do Ośrodka Elektronicznego GUS.

W zbiorach przekazywanych do GUS pomijane są rekordy zawierające określone błędy, dyskwalifikujące dalsze przetwarzanie w ramach Państwowej Statystyki Handlu



Rys. 3. Uproszczony schemat podziału systemu hz MHiPM na moduły eksploatacyjne



Zagranicznego. Rekordy takie, dopiero po usunięciu błędów przez PHZ, włączane są do zbiorów dla GUS w następnych cyklach przetwarzania. Natomiast błędy o mniejszej wadze, włączane są do zbiorów dla GUS maksymalnie w ciągu trzech kolejnych dekad, po czym system hz MHiPM — automatycznie nadając odpowiedni kod grupy błędów — eliminuje takie rekordy z dalszego przetwarzania, wymuszając niejako ich poprawienie przez PHZ.

Każdy rekord podlega symbolizacji w oparciu o zbiór jednostek gospodarki uspołecznionej REGON — zapewniający nadanie właściwego symbolu resortu i województwa, z uwzględnieniem automatycznego osymbolizowania zmian struktury organizacyjnej dostawców — w eksporcie oraz płatników — w imporcie. Z kolei następujące zbiory elementarne PHZ, gdzie każdy rekord odwzorowuje pojedynczą dostawę w ramach kontraktu, łączone są w rekordy sumaryczne wg klucza i struktury wymaganej przez GUS.

W zbiorach PHZ każdy rekord oznaczony jest miesiącem i dekadą wprowadzenia do zbioru, dlatego też ustalono, że w momencie wykonania zbioru dla GUS (w rejestrze obsługującym) następuje automatyczna zmiana dekady i miesiąca. Po wykonaniu zbiorów o strukturze zgodnej z wymaganiami GUS dla sześciu PHZ, zapisywane są łączne zbiory na dwóch taśmach magnetycznych (dla eksportu i importu), które czwartego dnia po zakończeniu dekady przekazywane są do GUS. Taśmy są zapisane w kodzie EBCDIC, natomiast konwersja na kod ICL następuje w Ośrodku Elektronicznym GUS.

W podobny sposób za każdy miesiąc sprawozdawczy przekazywane są na taśmach magnetycznych do CIHZ zbiory danych dotyczące kosztów i rodzajów transportu w hz. Są to dane elementarne odnoszące się do każdej wysyłki, wg określonej przez CIHZ zawartości informacyjnej i struktury rekordu. Obejmują one: numer kontraktu i faktury, symbol towaru, wartość w cenach transakcyjnych, koszty transportu i ubezpieczenia, ciężar brutto, ilość, rodzaj frachtu, symbol przejścia granicznego, kraj dostawcy i odbiorcy itp. Opracowana w uzgodnieniu z CIHZ kontrola programowa uwzględnia badanie poprawności powyższych danych wg kryteriów, których nie uwzględnia kontrola zbiorów GUS.

Należy dodać, że szereg informacji emitowanych przez CIHZ dla Centrali MHZ oraz przedsiębiorstw hz powstaje w wyniku przetwarzania zbiorów z realizacji obrotów hz dostarczanych do CIHZ przez GUS na taśmie magne-

tycznej. Dlatego też w kontroli programowej systemu MHiPM uwzględniono szereg parametrów badających poprawność symboli i ich logicznych powiązań, przyjętych w systemie CIHZ, a nie objętych kontrolą GUS (np. symbol umowy handlowej i kontrahenta zagranicznego). W ten sposób następuje zintegrowanie działań różnych uczestników Państwowego Systemu Handlu Zagranicznego (SISHZ).

Schemat przetwarzania modułu zasilania z PHZ pokazany jest na rysunku 4.

#### Zasilanie z GUS i CIHZ

Zasilanie z GUS i CIHZ obejmuje narastające zbiory transakcyjne hz resortu hutnictwa i przemysłu maszynowego w pełnym zakresie oraz zbiory obowiązującej w hz bazy normatywnej (katalogi, symbole, indeksy itp.). Zbiory transakcyjne dostarczane są do CO HiPM w ustalonych cyklach (dekadowo lub miesięczne), natomiast zbiory katalogowe kopiowane są natychmiast po każdorazowej ich aktualizacji. Dla ułatwienia współpracy każdy ze zbiorów rejestrowany jest na oddzielnej taśmie o ustalonej nazwie i numerze; taśmy te po ich wykorzystaniu odsyłane są niezwłocznie do GUS i CIHZ, oczekując tam na kolejny zapis zbioru.

Zbiory pochodzące z GUS zapisane są w kodzie ICL, a z CIHZ — w kodzie EBCDIC. Struktura zbiorów i rekordów dostarczanych do CO HiPM jest identyczna jak zbiorów głównych tworzonych w GUS i CIHZ, natomiast właściwe przystosowanie tych zbiorów do potrzeb systemu hz MHiPM następuje w CO HiPM. Realizuje to program, który identyfikuje zbiór i dokonuje konwersji na kod EBCDIC, zmienia strukturę rekordów i formę zapisu, a następnie wprowadza przekształcony i zapisany sekwencyjnie zbiór na przydzieloną automatycznie przez system taśmę magnetyczną lub dysk. Program ten aktualizuje także zapisany indeksowo-sekwencyjnie zbiór katalogowy. Aktualizacja taka odbywa się poprzez wymianę całego podzbioru lub pojedynczych rekordów.

Otrzymywane przez IOPM z GUS i CIHZ zbiory zapisanych narastająco kontraktacji i realizacji obrotów hz resortu, a także zbiory bazy normatywnej służą do emisji informacji niezbędnych do sterowania procesami hz przez jednostki organizacyjne MHiPM.

#### SPRZĘT INFORMATYCZNY W JEDNOSTKACH WSPÓLDZIAŁAJĄCYCH

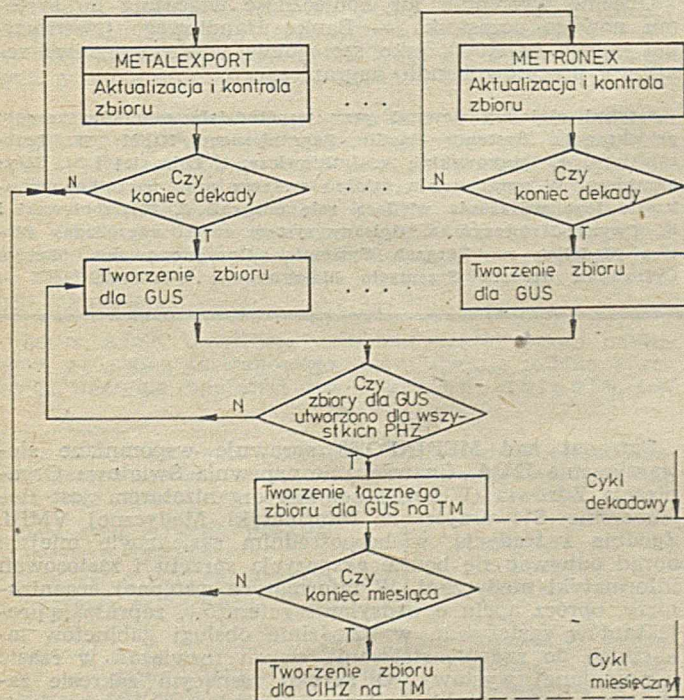
System informatyczny hz MHiPM eksploatowany jest przy wykorzystaniu istniejącego już (u jego uczestników) sprzętu komputerowego, co oznacza, że system ten nie powodował nowych wydatków inwestycyjnych. W miarę pojawiania się nowych komputerów następuje na nich wdrażanie odpowiednich fragmentów przetwarzania, z zachowaniem ustalonej jednolitej struktury i formy zapisu zbiorów na taśmach magnetycznych.

Większość procesów przetwarzania odbywa się w CO HiPM, wyposażonym w komputer IBM 370/145 z pamięcią operacyjną o pojemności 1 MB i systemem operacyjnym OS/VS1. Komputer ten jest wyposażony w pamięci dyskowe o pojemności 100 MB i 30 MB oraz pamięci taśmowe o gęstości zapisu 1600 bpi i 800/1600 bpi. W sieci teleprzetwarzania zainstalowane są u użytkowników m.in.: inteligentne terminale IBM 3735, monitory ekranowe IBM 3277 z drukarkami IBM 3284 lub MERA 7910 oraz wielostanowiskowa stacja terminalowa IBM 3790, wyposażona w pamięć dyskową, monitory ekranowe i drukarki.

W Ośrodku Elektronicznym GUS państwowy system informatyczny statystyki hz realizowany jest na komputerze ICL 1903 A z pamięcią operacyjną 128 K słów i systemem operacyjnym GEORGE 3 lub na komputerze ODRA 1305 z identyczną pojemnością pamięci i systemem operacyjnym. Komputery te wyposażone są w pamięci dyskowe o pojemności 8 i 60 M znaków.

CIHZ jest wyposażony w komputer IBM 360/50 z pamięcią operacyjną 512 KB, systemem operacyjnym OS/MFT, pamięciami dyskowymi o pojemności 30 MB oraz pamięcią taśmową o gęstości zapisu 800/1600 bpi. Część przetwarzania odbywa się na minikomputerze RC 5500 z pamięcią operacyjną 64 KB i systemem operacyjnym DOS oraz pamięciami dyskowymi o pojemności 25 MB i taśmowymi o gęstości zapisu 800 bpi.

Wyposażenie sprzętowe PHZ jest bardzo zróżnicowane. Posiadany sprzęt służy bądź tylko do rejestracji danych i przekazywania ich drogą teletransmisji do CO IOPM, bądź też — do re-



Rys. 4. Schemat przetwarzania modułu: Zasilanie z PHZ. Podano przykładowy schemat dla METAEXPORTU i METRONEXU, dla pozostałych PHZ (BUMAR, POLMOT, PEZETEL, UNITRA) odpowiednie fragmenty schematu są identyczne



alizacji własnych obiektowych systemów informatycznych, w tym również do przygotowania na taśmie magnetycznej zbiorów przeznaczonych dla CO HiPM.

— W PHZ PEZETEL i BUMAR zainstalowane są terminale typu IBM 3735, służące do rejestracji danych, emisji faktur statystycznych oraz transmisji danych do komputera IBM 370/145 w CO HiPM. Przetwarzanie danych na potrzeby tych PHZ odbywa się w CO HiPM w oparciu o prowadzone dla nich zbiory.

— W PHZ UNITRA zainstalowana jest wielostanowiskowa stacja terminalowa IBM 3790 o możliwościach zbliżonych do mini-komputera. Służy ona do rejestracji danych pod kontrolą programową, wstępnego ich przetwarzania, emisji faktur oraz transmisji danych do CO HiPM.

— W PHZ POLMOT zainstalowane są dwa terminale IBM 3735 oraz — jako sprzęt podstawowy — komputer HONEYWELL-BULL 61/60.2 z pamięcią operacyjną 96 KB, systemem operacyjnym GCOS 64 NC, monitorami ekranowymi z drukarkami, pamięciami dyskowymi o pojemności 30 MB oraz taśmowymi o gęstości zapisu 800 bpi.

— W PHZ METRONEX zainstalowany jest minikomputer RC 5500 z pamięcią operacyjną 64 KB i systemem operacyjnym DOS, pamięciami dyskowymi o pojemności 25 MB, taśmowymi o gęstości zapisu 800 bpi oraz monitorami ekranowymi.

— W PHZ METAEXPORT eksploatowany jest dwuprocessorowy minikomputer RC 6600 z pamięcią operacyjną 5 x 124 KB i systemem DOS/ARC, monitorami ekranowymi oraz pamięciami dyskowymi o pojemności 25 MB i taśmowymi o gęstości zapisu 800 bpi. Z powyższego wynika, że METAEXPORT na tle pozostałych PHZ dysponuje sprzętem o największym potencjale obliczeniowym.

Pomimo tak zróżnicowanego sprzętu — dzięki właściwej organizacji i technologii przetwarzania, a zwłaszcza stosowanym rozwiązaniom konwersji danych — system hz MHiPM funkcjonuje sprawnie, eliminując konieczność powtórzonego lub wielokrotnego rejestrowania dokumentów źródłowych.

#### ORGANIZACYJNO-PRAWNE ZABEZPIECZENIE WSPÓLDZIAŁANIA

Wdrażając system hz MHiPM, szczególną uwagę zwrócono na organizacyjno-prawną regulację przedsięwzięcia na styku MHiPM z GUS. Całość tej problematyki reguluje porozumienie o współpracy pomiędzy MHiPM a GUS, precyzujące m.in.:

- zasady organizacyjne współdziałania
- zakresy odpowiedzialności jednostek organizacyjnych GUS i MHiPM w realizacji porozumienia
- zasady finansowe współpracy.

W ramach ustaleń szczegółowych opracowana została precyzyjna instrukcja współpracy pomiędzy Centrum Obliczeniowym HiPM a Ośrodkiem Elektronicznym GUS, a także instrukcja współpracy PHZ z IOPM. Odpowiedzialność za jakość i terminowość tworzonych i dostarczanych zbiorów spoczywa zawsze na tym uczestniku systemu, który tworzy zbiór źródłowy, a więc za poprawność, kompletność i terminowość rejestracji danych elementarnych, związanych z realizacją obrotów, odpowiedzialne są poszczególne PHZ. Za poprawność parametrów kontrolnych odpowiada natomiast IOPM.

Z kolei jeśli GUS lub CIHZ nie dostarczą do CO HiPM w ustalonym terminie zaktualizowanych wersji zbiorów katalogowych, wówczas za spowodowane wskutek tej sytuacji błędy nie odpowiadają ani PHZ, ani IOPM.

Pisemnymi ustaleniami nie została objęta współpraca z Centrum Informatyki Handlu Zagranicznego. Wobec wciąż wzrastającej roli CIHZ w realizacji systemu państwowego, dotychczasowe unormowane tradycją współdziałanie okazuje się w szeregu przypadkach niewystarczające. Dlatego też w przygotowaniu znajduje się odpowiednie porozumienie pomiędzy CIHZ a IOPM.

\* \* \*

Dotychczasowa realizacja opisanego systemu, traktowanego jako integralny element systemu ogólnokrajowego, spowodowała:

- rozbudowanie jednolitej ogólnokrajowej bazy danych o obrotach hz, wykorzystywanej przez różne ogniwa zarządzania — zarówno w jednostkach organizacyjnych MHiPM, jak i w administracji centralnej
- spójność informacyjną oraz stosowanie jednolitej bazy normalatywnej pojęć, kategorii ekonomicznych i technicznych oraz innych symboli i indeksów opisujących obroty hz
- jednolitą, skoordynowaną kontrolę danych przekazywanych do systemu
- powszechne przekazywanie danych pomiędzy uczestnikami systemu na maszynowych nośnikach informacji (głównie na taśmach magnetycznych i drogą teletransmisji), przy przestrzeganiu zasady jednokrotnej rejestracji danych źródłowych
- wytworzenie się specjalizacji poszczególnych uczestników eksploatacji systemu; nastąpił wyraźny podział zadań i odpowiedzialności za wyniki
- powstanie podstawowego ogniwa w łańcuchu przetwarzania, w oparciu o które rozwinęły się obiektowe systemy informatyczne hz w różnych jednostkach resortu HiPM
- ugruntowanie przekonania kadry kierowniczej i personelu wykonawczego poszczególnych jednostek gospodarczych o celowości i racjonalności stosowania technik komputerowych do kierowania działalnością hz.

Obecnie występuje już konieczność włączenia do systemu nowego uczestnika — Banku Handlowego (ewentualnie innych banków), jako szczególnie ważnego ogniwa realizacji procesów handlu zagranicznego.

Artykuł niniejszy powstał przy współudziale członków zespołu projektowego Systemu Handlu Zagranicznego IOPM: W. Brzezińskiego, A. Dankowskiej, A. Jutkiewicza, I. Połaskiej i M. Struburskiego. W opracowaniu założeń systemu oraz początkowym okresie jego wdrażania wiodącą rolę odegrali J. Wierzbowski i K. Cwynar-Mrówczyńska. Opisany system został nagrodzony Złotym Medalem na Targach Systemów Oprogramowania Maszyn Cyfrowych Resortu Przemysłu Maszynowego SOFTARG'79.

## KONFERENCJE

### MEDINFO'83

Kongresy MEDINFO, poświęcone zastosowaniom informatyki w medycynie, zostały zainicjowane w 1974 r. przez Międzynarodowe Stowarzyszenie Informatyki Medycznej IMIA (International Medical Informatics Association). Stowarzyszenie to stanowi tzw. specjalistyczną grupę zainteresowań w ramach Międzynarodowej Federacji Przetwarzania informacji IFIP. W konsekwencji powyższego powiązania organizacyjnego, kongresy MEDINFO odbywały się równocześnie ze światowymi kongresami IFIP, a więc w cyklu trzyletnim (1974 Sztokholm, 1977 Toronto, 1980 Tokio). W roku 1983 po raz pierwszy następuje rozdzielenie miejsca i terminu obu kongresów: MEDINFO'83 odbędzie się nie we wrześniu w Paryżu, lecz w dniach 22—27 sierpnia w Amsterdamie.

Patronat nad MEDINFO'83 sprawuje wspomniane stowarzyszenie IMIA, finansowanie zapewnia Światowa Organizacja Zdrowia (WHO), natomiast organizatorem jest Holenderskie Stowarzyszenie Informatyki Medycznej VMBI. Zgodnie z tradycją, w bezpośrednim sąsiedztwie miejsca obrad odbywać się będzie ekspozycja sprzętu i zastosowań informatyki medycznej. W informatyce wstępnej organizatorzy, oprócz apelu o nadsyłanie referatów, zapraszają projektantów zastosowań w dziedzinie obsługi gabinetów lekarskich do zademonstrowania swych rozwiązań w czasie wspomnianej wystawy. O szybko rosnącym zakresie zastosowań informatyki medycznej świadczy ogłoszony program kongresu obejmujący aż 23 obszary tematyczne. Opłata za uczestnictwo została ustalona na poziomie 500 guldenów holenderskich (ok. 180 \$). Bliższych informacji udziela MEDINFO'83 Organizing Committee, Enschedepad 41, NL-1324 GB Almere-Stad, The Netherlands.



# AESN – pakiet programów do generowania wersji nakładanych programów fortranowych dla ODRY 1305

W przypadku dużych programów nie mieszczących się w pamięci operacyjnej, systemy kompilacji większości maszyn, wyposażonych w masową pamięć zewnętrzną, umożliwiają organizację dowolnego programu jako programu nakładanego. W tym celu program należy podzielić na jednostkę stałą i dowolną liczbę jednostek nakładanych.

Przy użyciu języka FORTRAN jednostka stała zawiera segment MASTER oraz może zawierać dowolne inne segmenty programu. W trakcie wykonywania znajduje się ona stale w określonym obszarze pamięci operacyjnej zwanym polem stałym. Pozostałe jednostki nakładania zawierają segmenty specyfikowane przez programistę i są przyrządowane określonym, innym obszarom pamięci operacyjnej zwanym polami nakładania.

W dowolnym momencie przebiegu programu każde z pól może zawierać tylko jedną z przypisanych mu jednostek. Aby nakładanie miało sens, każdemu polu nakładania powinny być przypisane co najmniej dwie jednostki nakładania. Wszystkie jednostki, z wyjątkiem jednostki stałej, znajdują się w pamięci zewnętrznej. Każda z nich, w przypadku wywołania segmentu nieobecnego w pamięci operacyjnej, jest stamtąd sprowadzana i umieszczana w przyporządkowanym jej polu nakładania, którego poprzednia wartość jest jednocześnie niszczone. Każde z pól nakładania musi mieć taką wielkość (faktycznie dotyczy to każdego podobszaru<sup>1)</sup> pola), aby mogło pomieścić największą z przypisanych mu jednostek.

Metoda uzyskiwania wersji nakładanych programów, w oparciu o którą zrealizowano pakiet AESN, może znaleźć zastosowanie dla dowolnego języka programowania i dowolnego typu komputera, o ile tylko ich systemy kompilacji w ogóle umożliwiają nakładanie.

Pakiet programów do generowania wersji nakładanych programów fortranowych został wykonany pod kierunkiem autora przez zespół informatyków w ośrodku komputerowym Biura Projektów i Realizacji Inwestycji Przemysłu Syntezy Chemicznej „PROSYNCHEM” w Gliwicach. Pakiet AESN został zrealizowany przy użyciu komputera ODRY 1305 działającego pod kontrolą systemu operacyjnego GEORGE-3. Jest on także przeznaczony do pracy na tym komputerze.

Pakiet AESN umożliwia uzyskanie takiej wersji nakładanej programu fortranowego, która spełnia zadane z góry ograniczenia wielkości tzw. pamięci dolnej<sup>2)</sup>, pa-

mieci zajmowanej przez instrukcje oraz pamięci całkowitej, zapewniając jednocześnie możliwie małą liczbę transferów jednostek nakładania z pamięci zewnętrznej do operacyjnej.

## POTRZEBA OPRACOWANIA PAKIETU

Biorąc pod uwagę aktualną sytuację gospodarczą kraju należy liczyć się z koniecznością wieloletniego jeszcze eksploataowania stosunkowo dużej liczby komputerów ODRY 1305. Są one dość powszechnie wykorzystywane w ośrodkach realizujących obliczenia inżynierskie i naukowo-badawcze. Podstawowym językiem programowania tego typu obliczeń, jest w tych ośrodkach język FORTRAN 1900.

Przy tworzeniu większych programów fortranowych dla ODRY 1305 natrafiono na szereg problemów utrudniających (lub wręcz uniemożliwiających) ich eksploatację. Do najważniejszych można zaliczyć:

- przekroczenie limitu pamięci dolnej
- przekraczanie granicy pamięci zajmowanej przez instrukcje programu, poza którą niemożliwe jest m.in. wykorzystywanie części oprogramowania standardowego ODRY 1305
- osiągnięcie bardzo dużych rozmiarów pamięci całkowitej.

Problemy te dały się mocno we znaki w trakcie realizacji i rozwoju Systemu Informatycznego Biura Projektów Przemysłu Chemicznego „CHEMOPRONET” (w latach 1974–1978 — problem resortowy MPChem nr 136, od 1979 r. — problem węzłowy 03.7. cz. III).

Dotychczasowa, żmudna praktyka „ręcznego” tworzenia wersji nakładanych okazała się zbyt kosztowna i mało efektywna:

— żmudna, gdyż polegała na konieczności analizowania wzajemnych powiązań wszystkich segmentów programu, a także szeregu innych informacji związanych z tymi segmentami; w większych programach liczba segmentów wynosi zwykle kilkadziesiąt, a nierzadko znacznie więcej,  
— zbyt kosztowna, gdyż sprawdzenie przydatności wersji nakładanej mogło się odbyć dopiero po jej kompilacji,  
— mało efektywna, gdyż nie sposób „ręcznie” przeanalizować takie zagadnienie w należyty sposób.

Najczęściej uzyskuje się tą drogą wersje nakładane o niepotrzebnie zaniżonej wielkości pamięci całkowitej, co zwykle powiększa liczbę transferów jednostek nakładania z zewnętrznej do operacyjnej pamięci komputera i w konsekwencji niepotrzebnie wydłuża rzeczywisty czas realizacji zadań. Jest to szczególnie uciążliwe w programach konwersacyjnych.

## ZADANIA I MOŻLIWOŚCI PAKIETU

Podstawowym zadaniem pakietu AESN jest generowanie wersji nakładanej programu fortranowego, spełniającej wymagania dotyczące wielkości pamięci. Pośrednio zaś pakiet stwarza możliwość:

- uzyskania odpowiedzi, czy dany program w ogóle może być eksploatowany
- wygenerowania fragmentu czołówki do kompilacji programu związanej z wierszami sterującymi OVERLAY

<sup>1)</sup> Dla maszyn cyfrowych serii ODRY 1300 struktura jednostki nakładania jest podobna do struktury pamięci programu. Jednostka nakładania zawiera następujące podobszary:

— stałe dolne (LOWER PRESET)	} podobszary nakładane jednostek
— literale (LITERALS)	
— stałe górne (UPPER PRESET)	
— instrukcja (PROGRAM)	
— zmienna dolna (LOWER VARIABLE)	} podobszary nienakładane jednostek (umieszczone w polu stałym)
— zmienna górna (UPPER VARIABLE)	
— obszary COMMON	

<sup>2)</sup> Dla uproszczenia przyjęto dla łącznej wielkości podobszarów LOWER PRESET, LITERALS i LOWER VARIABLE nazwę pamięć dolna. Pamięć dolna nakładana to suma wielkości LOWER PRESET i LITERALS.



- wskazania tych segmentów, które należy ponownie skompilować w odpowiednim trybie adresowania i skoków.

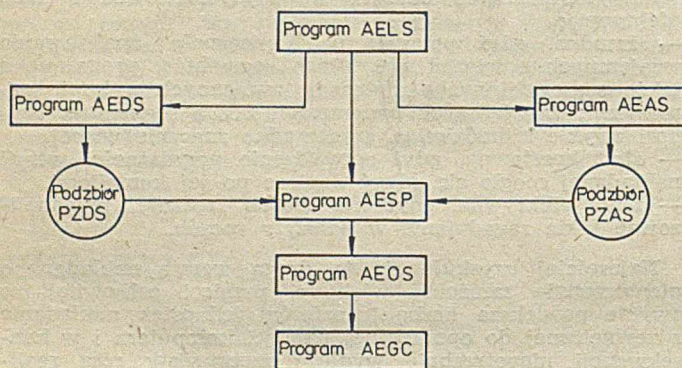
Uzupełniającym zadaniem pakietu AESN jest zakładanie i aktualizacja zbioru informacji o segmentach grup tematycznych oprogramowania. Na informacje te składają się:

- unikalny numer i nazwa segmentu
- rodzaj segmentu: nakładany lub nie nakładany
- wielkości pamięci całkowitej segmentu oraz jej części nakładanej
- wielkości pamięci dolnej segmentu oraz jej części nakładanej
- wielkość pamięci rozkazowej segmentu
- tryb adresowania: rozszerzony (extended) lub zwarty (compact)
- tryb skoków: rozszerzony (extended) lub zwarty (normal)
- liczba i numery segmentów wywoływanych w danym segmencie.

## BUDOWA I DZIAŁANIE PAKIETU

Pakiet programów AESN składa się z następujących elementów (rys.):

- programu AELS — ustalania listy segmentów i ich wzajemnych powiązań na drodze analizowania postaci źródłowej realizowanego programu fortranowego
- zbioru informacji o segmentach ZIS-X grupy tematycznej oprogramowania fortranowego; jest on złożony z dwóch podzbiorów:
  - PZDS, danych o segmentach
  - PZAS, adresów danych i wywołań segmentów
- programu AEDS tworzenia i aktualizacji podzbioru danych o segmentach
- programu AEAS — tworzenia i aktualizacji podzbioru adresów danych i wywołań segmentów
- programu AESP — kompletowania informacji o segmentach wykonywanego programu fortranowego
- programu AEOS — generowania struktury nakładanej wykonywanego programu fortranowego
- programu AEGC — generowania wierszy sterujących kompilacją wersji nakładanej wykonanego programu fortranowego
- makra MAESN — organizującego pracę całego pakietu w komputerze ODRA-1305 pod kontrolą systemu operacyjnego GEORGE-3.



Rys. Budowa i ogólny schemat działania pakietu AESN

Wyróżnić można trzy główne tryby działania pakietu:

**pełny** — w trakcie którego przed wygenerowaniem wersji nakładanej jest tworzony lub aktualizowany zbiór ZIS-X

**właściwy** — generujący wersję nakładaną w oparciu o kompletny zbiór ZIS-X

**przygotowawczy** — którego wyłącznym celem jest skompletowanie zbioru informacji o segmentach ZIS-X na drodze jego utworzenia lub aktualizacji.

## ZARYS METODY

Przetworzony na wersję nakładaną program fortranowy jest reprezentowany przez kwadratową macierz binarną o wielkości równej liczbie segmentów programu. Występowanie jedynek w *i*-tym wierszu i *k*-tej kolumnie oznacza wywołanie segmentu *k*-tego w *i*-tym. Początkowo zawartość tej macierzy, nazywanej często macierzą przyległości, odpowiada ściśle liście następników.

Po posortowaniu topologicznym segmentów, macierz przyległości jest przekształcana w tzw. macierz dostępności, która zawiera *explicite* informacje o powiązaniach nawet odległych od siebie (powiązanych poprzez ciąg pośrednich wywołań) segmentów. Algorytm takiego przekształcenia (domknięcia tranzytywnego) można znaleźć np. w [1].

Dla tak przygotowanej macierzy, uwzględniając informacje o wielkości pamięci segmentów (z rozbićm na podobszary) oraz aktualnie dostępnej pamięci operacyjnej komputera, stosuje się tzw. algorytm podziału [4]. Jego konstrukcja jest oparta na spostrzeżeniu, że podział macierzy na niezależne od siebie podmacierze tworzy gotowe jednostki nakładania, które mogą być przypisane temu samemu polu nakładania. Po dokonaniu podziału sprawdza się, czy suma pamięci wchodzących w skład największej (w sensie zajmowania pamięci) podmacierzy, po dodaniu do wielkości pamięci wszystkich wcześniej utworzonych pól nakładania oraz obszaru stałego, mieści się w dostępnej pamięci operacyjnej. Jeśli się nie mieści — tworzone jest nowe pole nakładania, a do przypisywanych mu jednostek trafiają te segmenty, których usunięcie z podmacierzy umożliwi ich rozbićm na kolejne, mniejsze podmacierze.

Dzielenia zaprzestaje się wtedy, gdy sprawdzenie da wynik pozytywny, względnie, gdy dalszy podział staje się niemożliwy. Ta ostatnia ewentualność pojawia się wówczas, gdy największa z podmacierzy jest jednoelementowa, względnie wtedy, gdy zawiera same jedynek — co oznacza, że powiązane są ze sobą wszystkie segmenty. Ewentualność ta jest oczywiście równoznaczna z nierozwiązalnością problemu dla zadanej, dostępnej pamięci komputera.

Algorytm jest realizowany kolejno dla poszczególnych typów obszarów pamięci zajmowanych przez segmenty (tj. pamięci dolnej, zajmowanej przez instrukcje oraz całkowitej) i ich ograniczeń określających dopuszczalną wielkość tych obszarów w szukanej wersji nakładanej. Po każdej kolejnej zmianie typu rozpatrywanego obszaru pamięci nie analizuje się macierzy od początku, lecz kontuuje podział, poczynając od podziału zastanego.

Metoda, według której realizowany jest algorytm, ma pewne dodatkowe, korzystne właściwości, a mianowicie zapewnia, że segmenty blisko ze sobą powiązane (tj. bezpośrednio lub poprzez niewielki ciąg wywołań), wraz z całymi podmacierzami, do których należą, trafiają — o ile tylko ograniczenia pozwolą — do tej samej jednostki nakładania. Tym samym wymusza się, by liczba transferów jednostek nakładania z pamięci zewnętrznej do pamięci operacyjnej była możliwie mała.

## GŁÓWNE CECHY METODY

Metoda [4] poszukiwania struktury nakładanej programu fortranowego, zastosowana w programie AEOS, charakteryzuje się następującymi własnościami.

- Podstawowym kierunkiem poszukiwań jest zmniejszanie (do osiągnięcia oczekiwanej wielkości) obszaru pamięci operacyjnej — zajmowanej przez program nakładany — przy spełnieniu ograniczeń związanych z jego pamięcią dolną i rozkazową.

- Minimalizowana jest *implicite* (w wyniku immanentnych właściwości metody) liczba transferów jednostek nakładania z pamięci zewnętrznej do operacyjnej.

- Nie jest analizowana charakterystyka realizacyjna programu, związana np. z liczbą iteracji w pętlach obejmujących całe segmenty.



Pakiet AESN stanowi narzędzie szczególnie pomocne w pokonywaniu trudności związanych z tworzeniem wersji nakładania programów i systemów inżynierskich zaprogramowanych w języku FORTRAN-1900 na komputerze Odra-1305. Można wyróżnić trzy zasadnicze sposoby jego użytkowania:

doraźny — inicjowanie pracy pakietu w trybie pełnym organizuje programista (użytkownik) programu fortranowego

automatyczny — inicjowanie pracy pakietu w trybie pełnym lub właściwym organizuje system inżynierski generujący (lub zawierający) postać źródłową programu fortranowego

półautomatyczny — inicjowanie pracy pakietu w trybie przygotowawczym organizuje system inżynierski generujący (lub zawierający) postać źródłową zestawów procedur (podprogramów) składających się na programy do rozwiązywania konkretnych zadań inżynierskich, natomiast inicjowanie pracy pakietu w trybie właściwym organizuje programista (użytkownik) odpowiedniego programu fortranowego.

Automatyczny sposób wykorzystania pakietu programów AESN został wdrożony w systemie do bilansowania masowego i energetycznego złożonych instalacji chemicznych PRODAS-80 [3], który na drodze formalnej analizy zadania bilansowego ustala optymalną kolejność obliczeń i dla ich realizacji generuje odpowiedni program fortranowy.

Półautomatyczny sposób wykorzystania pakietu został wdrożony w Programie Sterującym Współdziałaniem Banku Danych Fizykochemicznych Mediów Technologicznych z oprogramowaniem inżynierskim [2], który w oparciu o charakterystykę fizykochemiczną strumieni technologicznych twórcy w języku FORTRAN-1900 odpowiedni zestaw procedur sterujących i organizujących obliczenie własności fizykochemicznych w programie inżynierskim.

Wdrożenie pakietu programów AESN w ośrodku obliczeniowym Biura Projektów i Realizacji Inwestycji Przemysłu Syntezy Chemicznej PROSYNCHEM w Gliwicach odczuwalnie ułatwiło pracę programistom i jednocześnie wyraźnie zmniejszyło koszty tworzenia i eksploatacji wersji nakładanych fortranowych programów inżynierskich.

#### LITERATURA

- [1] Deo N.: Teoria grafów i jej zastosowania w technice i informatyce. BNI, Warszawa, 1980
- [2] Krop E., Polak M., Krzyżanowska A. i inni: Program sterujący (wersja taśmowa) współdziałaniem zbioru H-ZWF „Własności fizykochemiczne mediów technologicznych z oprogramowaniem inżynierskim”. EMC Odra-1305, PROSYNCHEM, Gliwice, 1977—1981
- [3] Krop E., Polak M., Rakowski L.: Obliczanie bilansów materiałowych i energetycznych złożonych instalacji chemicznych przy pomocy systemu PRODAS-80, zastosowanie elektronicznych maszyn cyfrowych w pracach inżynierskich. NOT, Katowice, 1981
- [4] Rakowski L.: Metoda ważonego podziału. Praca przygotowywana do druku.

## ALGORYTMY

# Sortowanie przez podział dystrybucyjny

Opis problemu i motywację wyjątkowo zastąpimy krótkim wprowadzeniem. W numerze 4—5/82 zaprezentowane zostały dwa algorytmy porównawcze sortowania wewnętrzne. Obecnie przedstawimy inną metodę sortowania — tzw. sortowanie przez rozsiew (zwane również dystrybucyjnym). W algorytmach porównawczych położenie elementu w ciągu uporządkowanym jest ustalane przez porównanie z innymi elementami; natomiast w algorytmach sortujących przez rozsiew — wartość elementu decyduje o jego położeniu w wektorze uporządkowanym. Rozsiew dokonywany jest przez przekształcenie klucza (lub jego części) w liczbę całkowitą, określającą w pewien sposób położenie elementu. Wymaga się więc, aby klucze były bądź liczbowe, bądź łatwo dające się przetworzyć lub interpretować jako liczbowe. Ograniczenie to dotyczy prawie wszystkich metod, w szczególności zaprezentowanego tu algorytmu sortowania przez podział dystrybucyjny (przedstawionego po raz pierwszy przez W. Dobosiewicza w jego pracy doktorskiej, Uniwersytet Warszawski, Instytut Informatyki, 1979).

Algorytmy sortujące przez rozsiew wymagają znacznej pamięci dodatkowej — zwykle proporcjonalnej do długości ciągu. Jednakże algorytm W. Dobosiewicza jest szczególnie atrakcyjny, gdyż za cenę zwiększenia pamięci uzyskujemy bardzo istotne przyspieszenie działania. Wystarczy tu wspomnieć, że DPS działa w średnim przypadku w czasie proporcjonalnym do ilości porządkowanych elementów ( $n$ ), zaś tylko w przypadku najgorszym czas ten zwiększa się do  $c \cdot n \cdot \log(n)$ .

#### Opis algorytmu

Ciąg wejściowy długości  $n$  dzielony jest, na podciągi  $G_1, \dots, G_n$  — takie, w których element podciągu  $G_i$  jest mniejszy od każdego elementu podciągu  $G_{i+1}$  (dla dowol-

nego  $i$  z zakresu  $1, \dots, n-1$ ). Takie samo postępowanie stosuje się następnie do każdego niepustego podciągu  $G_i$ , aż wektor stanie się całkowicie uporządkowany, tzn. gdy będzie  $n$  niepustych podciągów, z których każdy zawiera jeden element. Dokładniej, algorytm składa się z następujących kroków:

1. Znajdź najmniejszy, największy (procedura *minmax*) i środkowy (funkcja *mediana*) element porządkowanego ciągu (wartości tych elementów przechowywane są w zmiennych *min*, *max* i *median*).

2. Podziel każdy z odcinków  $\langle \text{min}, \text{median} \rangle$  i  $\langle \text{median}, \text{max} \rangle$  na  $n/2$  odcinków jednakowej długości.

3. Dokonaj rozsiewu ciągu porządkowanych elementów (procedura *dist*) w utworzone w kroku drugim przedziały. Element  $x$  leży w przedziale  $i$ -tym obliczonym według wzoru (funkcja *ival*):

$$i := \text{if } x \leq \text{median} \\ \text{then } (x - \text{min}) / (\text{median} - \text{min}) * (n-1)/2 + 1 \\ \text{else } (x - \text{median}) / (\text{max} - \text{median}) * (n-1)/2 + (n+1)/2$$

Wszystkie elementy, które trafią do  $i$ -tego przedziału tworzą podciąg  $G_i$ .

4. Usuń wszystkie puste podciągi (procedura *compress*); wtedy łączna liczba podciągów nigdy nie przekroczy liczby porządkowanych elementów.

5. Dla każdego  $i$  takiego, dla którego  $G_i$  istnieje, jeśli długość  $G_i$  jest większa niż 1, powtórz rekurencyjnie kroki 1—5, traktując  $G_i$  jako podciąg do posortowania.

#### Struktura danych

Przedstawiony algorytm jest rekurencyjny (a założenia tego cyklu nie dopuszczają programów rekurencyjnych), konieczne było zatem umieszczenie w procedurze DPS sto-



su. Funkcję stosu pełni tablica  $L$  o elementach typu integer ( $L[i]=k$  wtedy i tylko wtedy, gdy podciąg  $G_i$  ma  $k$  elementów) zaś aktualna wartość zmiennej zero wskazuje na wierzchołek stosu.

### Opis procedury

Procedura DPS ma dwa parametry:

- tablicę liczb rzeczywistych  $A$  (ciąg do sortowania)
- liczbę całkowitą  $n$  określającą długość tablicy  $A$ .

Dodatkowego wyjaśnienia wymaga funkcja *mediana*. Otóż ze względu na rozmiar tekstu programu zastąpiliśmy właściwe poszukiwanie elementu środkowego prostym wyliczeniem średniej arytmetycznej *min* i *max*. Oczywiście ta wersja procedury DPS nie zapewnia uzyskiwania pesymistycznego czasu rzędu  $n \cdot \log(n)$ . Zamiast tu podanej funkcji *mediana* należy wykorzystać zamieszczoną w poprzednim numerze funkcję *VK*; otrzymamy wówczas:

```
real procedure mediana;
begin
  mediana := VK (A, low, up, 1+(up-low)div 2, zero, L)
end mediana;
```

A to treść procedury PDS:

```
procedure DPS(A,n); real array A; integer n;
comment sortuje tablicę A[1:n] metodą dystrybucyjną;
begin integer array L[1:n]; integer low; up, up1, zero, i, j;
  real min, max, median; boolean wh;
  real procedure mediana;
  comment wartością powinien być środkowy co do wielkości element tablicy A;
  begin mediana := (min+max) / 2 end mediana;
  procedure minmax;
  comment znajduje najmniejszą (min) i największą (max) spośród liczb
    A[low], A[low+1], ..., A[up];
  begin integer i;
    min := max := A[up];
    for i := low step 2 until up-1 do
      if A[i] < A[i+1] then
        begin if A[i] < min then min := A[i];
          if A[i+1] > max then max := A[i+1]
        end
      else begin if A[i] > max then max := A[i];
        if A[i+1] < min then min := A[i+1]
        end
      end minmax;
  integer procedure ival(r); real r;
  comment wartością jest numer przedziału, do którego "wpada" r;
  begin
    if r < median then
      if min = median then ival := 1
      else ival := entier((r-min)/(median-min) * (up-low)/2+1)
    else
      if max = median then ival := up-low+1
      else ival := -entier(-((r-median)/(max-median)+1) * (up-low)/2+1);
    ival := ival+zero
  end ival;
```

```
procedure distr;
comment rozsiewa elementy tablicy A;
begin integer ir, i, j, t; real r, s;
  ir := up+1;
  for i := low step 1 until up do
    begin ir := ir-1; r := A[ir]; t := ival(r); j := L[t];
      if j > ir then
        begin L[t] := j-1;
          while j < ir do
            begin s := A[j]; A[j] := r; r := s;
              t := ival(r); j := L[t]; L[t] := j-1
            end;
          A[j] := r
        end if;
      end for;
  end distr;

procedure compress;
comment odzyskuje miejsce w tablicy L;
begin integer i, j;
  j := zero+1;
  for i := zero+2 step 1 until up1 do
    if L[i] < L[i-1] then begin L[j] := L[i]; j := j+1 end;
    zero := j; L[j] := up
  end compress;
procedure nextpart;
comment znajduje kolejny nieposortowany przedział;
begin
  while L[zero] - L[zero-1] < 1 do zero := zero-1
end nextpart;

comment początek DPS;
if n > 1 then
  begin low := 1; up := n; L[-1] := -2; L[0] := zero; zero := 0; wh := true;
    while wh do
      begin minmax;
        if min = max then
          if zero < 1 then wh := false
          else begin zero := zero-1; nextpart end
        else
          begin median := mediana; up1 := zero+up-low+1;
            for i := zero+1 step 1 until up1 do L[i] := 0;
            for i := low step 1 until up do begin j := ival(A[i]); L[j] := L[j]+1 end;
            for i := zero+1 step 1 until up1 do L[i] := L[i]+L[i-1];
            distr; compress; nextpart
          end;
          if zero < 1 then wh := false
          else begin low := L[zero-1]+1; up := L[zero]; zero := zero-1 end
        end while;
      end if;
  end DPS;
```



## BASIC

# O pewnym rozwiązaniu układu równań liniowych

W odpowiedzi na propozycję zawartą w liście Jacka Żebrowskiego (por. INFORMATYKA 2—3/82) Danuta Kalinowska nadesłała do naszej rubryki dwie procedury napisane w BASICU. Wybraliśmy do zaprezentowania jedną z nich, naszym zdaniem interesującą, otwierając tym samym kącik procedur BASICOWYCH. Zachęcamy Czytelników do przedstawiania ciekawych rozwiązań problemów, z którymi spotykali się w praktyce.

Pomysł p. Kalinowskiej dotyczy rozwiązywania układów liniowych pewnej szczególnej postaci, odstaje więc nieco od naszej koncepcji prezentowania algorytmów nienumerycznych. Sądzymy jednak, iż kącik BASICOWY może zawierać również programy numeryczne, zwłaszcza takie, które mogą ułatwiać pracę użytkownikom systemów mikroprocesorowych oraz innych komputerów o ubogich zasobach sprzętowych. (Red.)

Należy rozwiązać układ równań liniowych stopnia  $2n$  ( $n \leq 40$ ) o macierzy postaci:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

gdzie  $A, D$  są macierzami diagonalnymi stopnia  $n$ , zaś  $C, B$  — macierzami pełnymi stopnia  $n$ .

## Motywacja

Program pozwala na rozwiązanie układu równań stopnia 80 nie mieszczący się w pamięci minikomputera WANG-2200 (cała pamięć wypełniona jest przez macierz kwadratową stopnia 63). Rozwiązania układu wyznaczane są z dużą dokładnością.

## Opis rozwiązania

Układ równań liniowych stopnia  $2n$  ( $n \leq 40$ ) postaci:

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \cdot \begin{bmatrix} E \\ F \end{bmatrix} = \begin{bmatrix} U \\ W \end{bmatrix}$$

rozwiązujemy poprzez odwracanie macierzy:

$$Q = \begin{bmatrix} A & B \\ C & D \end{bmatrix}$$

$$Q^{-1} = \begin{bmatrix} K & L \\ M & N \end{bmatrix}$$

gdzie  $N = (D - CA^{-1}B)^{-1}$ ,  $M = -NCA^{-1}$ ,  $L = -A^{-1}BN$ ,  $K = A^{-1} - A^{-1}BM$ .

Stąd

$$E = A^{-1}U - A^{-1}B(MU + NW), F = MU + NW$$

Zatem algorytm wyznaczania  $E, F$  może być następujący:

o  $C := D - CA^{-1}B$ ;  $B := C^{-1}$ ; teraz zawartość  $B$  jest równa  $N$

- o  $X := BW$ ; teraz  $X = NW$
- o wprowadź macierz  $C$  z dysku do pamięci operacyjnej
- o  $B := NCA^{-1}$ ; teraz zawartość  $B$  jest równa  $-M$
- o  $E := BU$ ; teraz  $E = -MU$
- o  $F := -E + X$ ;  $F$  jest jednym z rozwiązań
- o  $E := -F$
- o wprowadź macierz  $B$  z dysku do pamięci operacyjnej
- o  $Z := A^{-1}BE$
- o  $E := A^{-1}U + Z$ .

Czas wyznaczania rozwiązań  $E$  i  $F$  według powyższego algorytmu dla macierzy stopnia  $n = 80$  wynosi ok. godziny na minikomputerze WANG-2200.

Algorytm realizujemy za pomocą dwóch programów:

- o **POCZATEK**, który buduje macierz układu oraz wektor prawych stron, przesyła na dysk macierze  $B, C$ , zaś wektory  $A, D$  (przekątne macierzy diagonalnych) i  $U, W$  (prawych stron) przechowuje w obszarze COMMON; następnie wprowadza do pamięci operacyjnej program **ROZ.UKL**, powiązany z nim obszarem COMMON.

```

4 REM * PROGRAM 'POCZATEK'
5 REM * BUDOWA UKŁADU ROWNAN
15 COM A(40), D(40), U(40), W(40), B(40,40), C(40,40), E(40), F(40), N, B7*, C7*
30 INPUT "PODAJ STOPNIEN MACIERZY (N<=80-LICZBA PARZYSTA)",N:N=N/2
40 MAT REDIM A(N), D(N), U(N), W(N), B(N,N), C(N,N), E(N), F(N)
50 INPUT "PODAJ NAZWE ZBIORU DLA MACIERZY B",B7*
60 INPUT "PODAJ NAZWE ZBIORU DLA MACIERZY C",C7*
101 REM * NALEZY UTWORZYC MACIERZ 4-BLOKOWA
102 REM * MACIERZE DIAGONALNE A, D - PODAC W POSTACI WEKTOROW A, D N-ELEMENTOWYCH
103 REM * BLOKI B, C - PODAC W POSTACI PEŁNYCH MACIERZY WYMIAROW N*N
104 REM * WEKTOR PRAWYCH STRON PODAC W WEKTORACH W1, W2 N-ELEMENTOWYCH
370 REM * ZAPIS NA DYSK MACIERZY B, C
375 I=INT(N*N/28)+4
380 DATA SAVE DC OPEN R1,B7*
390 DATA SAVE DC B( )
400 DATA SAVE DC END
420 DATA SAVE DC OPEN R1,C7*
430 DATA SAVE DC C( )
440 DATA SAVE DC END
460 MAT REDIM A(40), D(40), B(40,40), C(40,40), U(40), W(40), E(40), F(40)
470 LOAD DC F "ROZ.UKL" 20,500
485 REM * PROGRAM 'ROZ.UKL' ROZWIADUJE ZBUDOWANY POWYZEJ UKŁAD ROWNAN
500 STOP : END

● ROZ. UKŁ, który realizuje algorytm wyznaczania E i F.
30 REM * PROGRAM 'ROZ.UKL' POWIAZANY POPRZEZ COMMON Z 'POCZATEK'
45 DIM Z(40), X(40)
50 MAT REDIM A(N), D(N), B(N,N), C(N,N), U(N), W(N), Z(N), E(N), F(N), X(N)
120 FOR I=1 TO N:FOR J=1 TO N:Z=0:FOR K=1 TO N:Z=Z+C(I,K)/A(K)+B(K,J)+NEXT K:Z:Z: NEXT J
150 FOR J=1 TO N:C(I,J)=Z:Z: NEXT J:C(I,D)=D+C(I,D): NEXT I
170 MAT B=INV(C)
180 FOR I=1 TO N:Z=0:FOR J=1 TO N:Z=Z+B(I,J)*W(J):NEXT J
190 X(I)=Z:Z: NEXT I
200 DATA LOAD DC OPEN R C7*
205 DATA LOAD DC C( )
210 IF END THEN 220
220 FOR I=1 TO N:FOR J=1 TO N:Z=0:FOR K=1 TO N:Z=Z+B(I,K)*C(K,J)+NEXT K:Z:Z: NEXT J
:FOR J=1 TO N:B(I,J)=Z/J/A(I):NEXT J: NEXT I
230 FOR I=1 TO N:Z=0:FOR J=1 TO N:Z=Z+B(I,J)*U(J):NEXT J:E(I)=Z:Z: NEXT I
240 FOR I=1 TO N:F(I)=-E(I)+X(I):NEXT I
300 DATA LOAD DC OPEN R B7*
305 DATA LOAD DC B( )
310 IF END THEN 320
320 FOR I=1 TO N:E(I)=-F(I):NEXT I
330 FOR I=1 TO N:Z=0:FOR J=1 TO N:Z=Z+B(I,J)*E(J):NEXT J
340 Z(I)=Z/A(I):NEXT I
350 FOR I=1 TO N:E(I)=Z(I)+U(I):NEXT I
360 REM * WEKTORY E, F TWORZA ROZWIADANIE UKŁADU

```



## Zastosowania komputerów w przemyśle

W roku ubiegłym odbyły się dwie krajowe konferencje dotyczące przemysłowych zastosowań systemów komputerowych: „Zastosowanie komputerów w przemyśle” (Szczecin, 17—18. września) i „Aparatura systemu CAMAC w zastosowaniach przemysłowych” (Warszawa, 8—9 października). Referaty przedstawione na obu konferencjach stanowią wystarczającą podstawę do ogólnego spojrzenia na zagadnienia stosowania komputerów w przemyśle.

Zdobycie miarodajnych informacji i poddanie ich ocenie jest jednak dosyć trudne. Spośród 95 referatów bowiem zaledwie dwa pochodziły z zakładów przemysłowych nie produkujących środków informatyki (J. Zaremba, Inowrocławskie Zakłady Chemiczne; A. Nawrocki, Stocznia im. Lenina).

Dla uzyskania prawdziwego obrazu najlepiej będzie pominąć prace badawcze i projektowe, a zająć się rzeczywistymi zastosowaniami. A przy tym — ze względu na fakt, że ani sam produkowany lub używany sprzęt, ani samo oprogramowanie (podstawowe lub użytkowe) nie decydują o tym obrazie — nie uwzględniać typowego podziału na prace dotyczące sprzętu i oprogramowania.

### KOMPUTEROWE STEROWANIE OBIEKTÓW PRZEMYSŁOWYCH

Najbardziej oczekiwanym zastosowaniem komputerów (a ściślej — mikroprocesorów) jest wykorzystanie ich w wyrobach przemysłowych. Niestety — te czasy jeszcze dla nas nie nadeszły. Tylko w jednym przypadku (M. Kalus i in., Instytut Podstawowych Problemów Elektrotechniki PSl.) wyczułem intencje wyposażenia wyrobu w mikroprocesor (regulacja prędkości silnika prądu stałego). Tam gdzie jest to konieczne dla zwiększenia konkurencyjności wyrobów, np. na stawkach, zaledwie rozpoczyna się prace w tym kierunku (urządzenia antykolizyjne, urządzenia nawigacyjne, sterowanie siłowni, elektrowni okrętowych).

Warto zwrócić uwagę na szeroki zakres prac i wyniki komputeryzacji obiektów chemicznych. W kilku zakładach chemicznych (np. CHEMITEK-ELANA w Toruniu, Zakłady

Chemiczne POLICE) zrealizowano z powodzeniem różne formy komputerowego sterowania procesami, najczęściej przy użyciu importowanego sprzętu i własnego oprogramowania użytkowego. Zapewne jednym z czynników sprzyjających tu stosunkowo dużemu zaawansowaniu komputeryzacji jest istnienie resortowego ośrodka automatyzacji CHEMOAUTOMATYKA, z którego pochodzą autorzy czterech prezentowanych prac.

Ciekawym przykładem komputeryzacji procesu chemicznego jest sterowanie produkcją sody w Inowrocławskich Zakładach Chemicznych (H. Bieliński, Instytut Chemii Nieorganicznej, Gliwice; J. Zaremba, I. Z. Chem.). Jest to druga próba automatyzacji tego procesu. Pierwszą podjęto w latach siedemdziesiątych (w oparciu o urządzenia krajowe) w Janikowskich Zakładach Sodowych — w ramach prac koordynowanych przez Przemysłowy Instytut Automatyki i Pomiarów (PIAP). Instytut ten jest ośrodkiem o dużym dorobku w dziedzinie komputeryzacji sterowania obiektów przemysłowych; przykładem realizacji jest system centralnej rejestracji i przetwarzania danych w dużym obiekcie — cukrowni (J. Jurkowski, W. Kozłowski, W. Przerwa).

Również w innych gałęziach przemysłu, prowadzone są prace nad zastosowaniem komputerów do bezpośredniego nadzorowania lub sterowania obiektów, np. bloków energetycznych elektrowni w Połańcu (W. Kazienko i in., OBR Systemów Automatyki w Ostrowiu Wielkopolskim); sieci gazowniczej Górnośląskiego Okręgu Przemysłowego (J. Lipowski i in., Instytut Informatyki PSl.); procesu wielkopięcowego w Hucie im. Lenina (A. Englot, W. Hejmo, Instytut Elektrotechniki PKr.) czy linii technologicznej do produkcji tzw. szarpanki w Bielskiej Fabryce Maszyn Włókienniczych (R.N. Fila, A. J. Szymborski, Instytut Systemów Sterowania w Katowicach).

Pewnym zaskoczeniem były dla mnie aż trzy referaty dotyczące zastosowania komputerów w przemyśle włókienniczym (choć nie do sterowania). Na próżno zaś szukałem referatów związanych z przemysłem samochodowym, choć wiadomo, że importowane linie technologiczne są skomputeryzowane.

### NOWE ZAGADNIENIA W STEROWANIU KOMPUTEROWYM

Oprócz klasycznych problemów — doboru odpowiedniego sprzętu, oprogramowania i algorytmów sterowania, często podkreśla się obecnie ogromną rolę właściwej komunikacji operatora z procesem lub komputerem (E. Grabowski, J. Wyrowiński, P. Szymański, CHEMOAUTOMATYKA; W. Kozłowski, M. Kazienko i in.) oraz duże znaczenie struktury danych w sterowaniu (W. Bobrzyński, M. Zabińska, Instytut Informatyki AGH; J. Jurkowski, W. Przerwa, J. Lipowski i in.). Do bardzo istotnych zagadnień należy także niezawodność

oprogramowania, co zapewniano m.in. przez przechowywanie kopii stanu programów. Niezwykle ważne jest rozpoznanie znaczenia lokalnych sieci komputerowych w sterowaniu (R. Kwaśniewski, M. Lizurek, Instytut Informatyki PW; I. Józwiak, Instytut Cybernetyki Technicznej PWr.; M. Kacprzak, Instytut Maszyn Matematycznych w Warszawie; J. Lipowski i in.).

Z treści prezentowanych prac wynika, że w naszych warunkach bardzo skutecznym narzędziem automatyzacji obiektów przemysłowych są modularne systemy cyfrowe. Pełna normalizacja ma istotne znaczenie dla szybkiego i łatwego dobrego konfiguracji i skompletowania zestawu oraz dla natychmiastowej naprawialności sprzętu, cechującego się dużą zawadnością. Lektura referatów pozwala stwierdzić, że wiodącą rolę w przemyśle pełni system INTEL DIGIT-PI, opracowany w PIAP. Należy jednak przypuszczać, że już wkrótce zacnie z nim rywalizować system CAMAC (R. Zarzycki, MERA-PNEFAL; A. Dec, POLON). Niemniej — choć współpraca projektantów (Instytut Badań Jądrowych) i producentów bloków (POLON) — jest jak dotąd bardzo skuteczna — przykłady innych ośrodków (CHEMOAUTOMATYKA, PIAP) świadczą o tym, że tworzenie pełnych zestawów i kompleksowe podejście do zagadnień komputeryzacji obiektów przemysłowych jest realizowane najpełniej w jednostkach wyodrębnionych.

### ZASTOSOWANIE KOMPUTEROW W ZARZĄDZANIU PRZEDSIĘBIORSTWEM PRODUKCYJNYM

Zastosowanie komputera do sterowania obiektem w czasie rzeczywistym charakteryzuje się tym, że po uruchomieniu instalacji w zasadzie nie można komputera usunąć. W zarządzaniu natomiast rola komputera jest pod tym względem ograniczona, ponieważ nie wymusza on decyzji i jego informacje mogą być pominięte — jak się sądzi — bez szkody dla przedsiębiorstwa (nie trzeba chyba wyjaśniać jak szkodliwy jest to pogląd). Może dlatego nie znalazłem ani jednego referatu dotyczącego zrealizowania pełnego systemu komputeryzacji (lub komputerowego wspomaganie) zarządzania, choć zagadnieniem tym poświęcono wiele prac teoretycznych.

Przedstawiono — na przykład — bardzo ciekawy problem operatywnego sterowania produkcją (M. Zaborowski, Instytut Automatyki PSl.), polegający na minimalizacji odchyień wartości produkcji, spowodowanych takim zakłóceniami, jak:

- nierytmiczność dostaw surowców lub dostaw kooperacyjnych
- wahania zdolności produkcyjnej poszczególnych oddziałów (awarie, absencja pracowników, opóźnienia remontów, wyłączenia energii, itp.)
- zmiany planu zbytu, zakłócenia w ekspedycji, brak środków transportu itd.



Bardzo efektywnym, szczególnie w zagadnieniach z zakresu zarządzania jest komputerowe planowanie i organizowanie przeładunku kontenerów, któremu poświęcono dwa referaty (J. Dądela, Wyższa Szkoła Morska w Gdyni; K. Amborski i in., Instytut Sterowania i Elektroniki Przemysłowej PW).

Stwierdzono, że podstawowe znaczenie dla skuteczności zastosowania komputerów w zarządzaniu ma nie tyle staranne przygotowanie algorytmów, ile właściwy dobór i rozmieszczenie sprzętu komputerowego, służącego do wprowadzania i wyprowadzania informacji. Automatyzacja czynności wejściowych (A. Chowański, Instytut Komputerowych Systemów Automatyki i Pomiarów we Wrocławiu), wyjściowych (B. Czaiński, J. Peszat, Zakład Problemów Organizacji i Zarządzania PAN w Bytomiu) i zintegrowanie punktów wymiany informacji w lokalną sieć komputerową (R. Kwaśniewski, M. Lizurek) stanowią warunek sprawnego zarządzania z wykorzystaniem komputera. W ten sposób zapewnia się szybkość dostarczania i sprawdzalność poszczególnych informacji. Istotne jest także współdziałanie wszystkich komputerowych podsystemów składowych, tj. systemów: finansowo-księgowego, ewidencji pracowników, zapotrzebienia materiałowego, ewidencji maszyn i urządzeń oraz sterowania i kontroli produkcji (R. Kwaśniewski, M. Lizurek).

Wśród wielu przyczyn braku większego zainteresowania komputeryzacją ze strony przemysłu można wymienić (poza niewspółmiernie wysokimi cenami sprzętu i usług, a także złożonością komputerów) nieprzystosowanie zainstalowanych komputerów do komunikacji z przeciętnym użytkownikiem (np. mistrzem lub brygadziście).

#### PROJEKTOWANIE WSPOMAGANE KOMPUTEROWO

Automatyzacja projektowania inżynierskiego jest niezwykle ważną dziedziną zastosowań komputerów. Na konferencji w Szczecinie przedstawio-

no wiele prac na ten temat, dotyczących m.in. zastosowania komputerów w projektowaniu układów mechanicznych, pneumatycznych, elektronicznych, elektrycznych, a także — w projektowaniu statków czy odzieży.

Dla specjalistów, sesja dotycząca projektowania była z pewnością interesująca, jednak w porównaniu ze stanem techniki światowej prace te nie mają istotnego znaczenia. Dlatego nie będę ich omawiał. W tej dziedzinie do czołówki światowej brakuje nam co najmniej tyłu lat, ilu w produkcji mikroprocesorów. W Polsce prace nad sprzętem i oprogramowaniem przeznaczonym do projektowania nie wyszły, w najlepszym przypadku, poza fazę opracowań (S. Bonkiewicz-Sittauer, Instytut Maszyn Matematycznych; M. T. Jankowski, Instytut Technologii Elektronowej, CE-MI).

\* \* \*

Na podstawie przebiegu obrad obu konferencji mogę stwierdzić, że:

- upłyne dużo czasu zanim do wyrobów przemysłowych zaczniemy wprowadzać mikroprocesory (a jest to konieczność dziejowa)
- komputerowe sterowanie obiektów przemysłowych jest uwarunkowane niechęcią przemysłu do stosowania komputerów w procesie produkcyjnym, m.in. wskutek niedoskonałości samego sprzętu komputerowego i oprogramowania (brak elastyczności, wymienności i niezawodności)
- poziom techniki umożliwiającej komputerowe projektowanie wyrobów w różnych gałęziach przemysłu jest katastrofalnie niski
- stosowanie komputerów w zarządzaniu przedsiębiorstwem będzie miało szanse powodzenia dopiero po uporządkowaniu gospodarki, choć potencjalne możliwości takiego uporządkowania w skali przedsiębiorstwa tkwią właśnie w komputeryzacji różnych fragmentów procesu zarządzania.

JANUSZ ZALEWSKI

### Poszukiwanie literatury

Wszyscy wiemy, jak trudno jest dzisiaj dotrzeć do niezbędnej literatury fachowej. Z wielu powodów ostatnio bardzo już zmalał jej dopływ, a możliwe, że wobec narastających napięć w sytuacji międzynarodowej oraz konieczności dalszych ograniczeń dewizowych dopływ ten zmniejszać się będzie nadal. Trudno sobie jednak wyobrazić, by mógł on ustać zupełnie, dlatego apelujemy do przedstawicieli bibliotek, instytutów i ośrodków informatycznych, aby informowali nas o nowych nabytkach (gł. czasopiśmie informatycznych) i umożliwiali nam ich wykorzystanie na łamach. Chętnie publikowalibyśmy też informacje bibliotek, przedstawiające rodzaj i zakres tematyczny zbiorów oraz sposób dostępu do nich. (Red.)

## GIEŁDA INFORMACJI

Publikujemy bezpłatnie:

- krótkie prezentacje nowych, nie wykorzystanych dostatecznie systemów
- doniesienia o innowacjach programowych
- ogłoszenia dotyczące nowych rozwiązań technicznych sprzętu informatycznego
- indywidualne oferty informatyków poszukujących pracy oraz informacje o wolnych etatach
- ogłoszenia w sprawie szkoleń informatycznych
- inne informacje ułatwiające doradczą organizację informatyki w Polsce.

### Drukowanie Tabulogramów

Fabryka Samochodów Małolitrażowych Zakład nr 2 w Tychach oferuje zestaw programów ułatwiających drukowanie tabulogramów na komputerach Jednolitego Systemu, pracujących pod systemem operacyjnym OS:

- Program drukowania tabulogramów zapisanych uprzednio na taśmie magnetycznej, pozwalający na:
  - tworzenie dowolnej liczby kopii tego samego tabulogramu
  - drukowanie dowolnie wybranego fragmentu tabulogramu poprzez deklarację strony początkowej lub przedziału stron
  - restartowania wydruku od dowolnej strony.

Sterowanie pracą programu może odbywać się za pośrednictwem kart parametrycznych lub na bieżąco z konsoli operatorskiej. Program jest szczególnie przydatny w sytuacjach przerwań papieru przez drukarkę wierszową, dodrukowywania wybranych stron katalogów, itp.

• Program sporządzania skorowidza („mapy”) wydruków z koleiki wyjściowej zapisanej na taśmie magnetycznej. Program ten może być wykorzystany, gdy np. — wskutek całkowitej awarii drukarki wierszowej — systemowy program WTR zapisał kolejkę wyjściową na taśmę magnetyczną. Wówczas według sporządzonego najmniejszym programem skorowidza można selektywnie wydrukować żądany tabulogram pierwszym programem.

Blizszych informacji udzielają: mgr Piotr Piwko (tel. 27-94-47); mgr inż. Eugeniusz Szmajduch (tel. 27-94-49) — FSM Tychy, Zakład Informatyki ul. Oświęcimska 401.



## GIEŁDA INFORMACJI

### Gospodarka materiałowa

System opracowano w ZETO-Wrocław dla użytkowników MERY-9150, eksploatujących system „Gospodarka Materiałowa — KTM” (tego samego autorstwa) na ODRZE lub RIADZIE. Niezależnie od powiązań między obu systemami „Gospodarkę Materiałową” można eksploatować oddzielnie, jako niezależny, całościowy system ewidencji materiałowej.

Podstawową funkcją systemu jest:

- wprowadzanie dokumentów ze szczegółową kontrolą danych wejściowych (numeryczność pól, wartości graniczne, poprawność cyfry kontrolnej indeksu itp.)
- listowanie błędnych pozycji
- zakładanie i lub aktualizacja kartoteki stanów magazynowych materiałów i stanów przedmiotów nietrwałych u osób odpowiedzialnych
- możliwość natychmiastowego wyświetlenia na ekranie monitora żądanej pozycji kartoteki.

Stosując system „GM-KTM”. (SEECHECK) unika się podstawowych ograniczeń „GM-KTM” realizowanych przy wykorzystaniu dużego komputera, tj.: wykrywania błędów w trakcie realizacji obliczeń oraz uzyskiwania wyników z opóźnieniem. Dzięki stosowaniu systemu SEECHECK na komputer ODRA lub RIAD trafiają tylko dokumenty poprawne.

System może być stosowany przez każde przedsiębiorstwo posiadające MERE-9150. Może być on na żądanie użytkownika rozbudowany o dodatkowe zestawienia realizowane na MERZE-9150.

Zgłoszenia przyjmuje kierownik Działu Marketingu mgr Tadeusz Czerniewski; ZETO-Wrocław, ul. Ofiar Oświęcimskich 7/13, 50-069 Wrocław; tel. centr. 44-54-31 do 37, telex 0712533 ZETO PL.

### ITEM

System ten, opracowany przez ZETO-Lódź, usprawnia pracochłonne procesy projektowania tzw. map ekranowych na monitorach typu MERA-7900 lub podobnych — współpracujących z komputerami rodziny RIAD/IBM o minimum 512 KB pamięci operacyjnej. Warunkiem technicznym eksploatacji systemu ITEM jest możliwość pracy pod systemem operacyjnym OS w wersji wielozadaniowej oraz wykorzystanie

## Kadra kierownicza PGR o potrzebie zastosowań informatyki

Rolnictwo charakteryzuje się nieregularnością działania czynników produkcji, których czas występowania, siłę i kierunki działania trudno przewidzieć. Synchronizacja zjawisk rolniczych jest determinowana ich przyrodniczym charakterem, zależnością od przebiegu czynników klimatycznych, właściwości gleby, żywych organizmów roślinnych i zwierzęcych, sezonowością nakładów uwarunkowanych agrotechniką i vegetacją roślin. Powoduje to w odróżnieniu od innych gałęzi gospodarki (np. przemysłu) niemożliwość zautomatyzowania głównych procesów produkcji lub też istotne w tym trudności!).

Sytuacja społeczno-gospodarcza kraju spowodowała, że z dniem 1 lipca 1981 r. państwowe gospodarstwa rolne (PGR-y) przeszły na rozrachunek gospodarczy. Ta zasada zarządzania gospodarstwami przyznaje im dużą samodzielność prawną i ekonomiczną. Implikuje to konieczność wprowadzenia do praktyki gospodarczej zasady opłacalności produkcji. Wydaje się, że jednym z narzędzi usprawniających ten proces może być informatyka.

Dla zasięgnięcia opinii kadry kierowniczej PGR o potrzebie zastosowań informatyki w rolnictwie społecznym Pomorza Zachodniego przeprowadzono w roku akademickim 1981/1982 sondaż. Obejmował on studentów II roku Wydziału Rolniczego Studiów Stacjonarno-Zaocnych Akademii Rolniczej w Szczecinie. Wszyscy respondenci mieli opanowany materiał z zakresu podstaw informatyki i zastosowań metod matematycznych w praktyce rolniczej oraz legitymowali się długoletnim stażem pracy (od 4 do 15 lat) w rolnictwie i na stanowiskach kierowniczych w PGR. Sondaż przeprowadzono metodą anonimowej ankiety. W swoich wypowiedziach ankietowani podjęli siedem grup tematycznych, które wydawały się najbardziej istotne z punktu widzenia reprezentowanych przez nich przedsiębiorstw, mianowicie:

- terminu wprowadzenia informatyki

- modelu przedsiębiorstwa
- dziedziny zastosowań
- barier i ograniczeń
- sprzętu informatycznego
- przetwarzania i organizacji ośrodków obliczeniowych
- rodzaju systemów informatycznych.

Na trzydziestu respondentów — reprezentujących tyleż państwowych gospodarstw rolnych — 20% wypowiedziało się za wprowadzeniem informatyki już w chwili obecnej, 73% — za wprowadzeniem w okresie późniejszym, tj. po uporządkowaniu istniejącego stanu gospodarczego, natomiast 7% odpowiedzi było negatywnych. Respondenci pochodzili z przedsiębiorstw różnej wielkości (od 8 do 15 tys. h użytków rolnych) oraz o zróżnicowanym starcie finansowym.

Jeśli chodzi o model przedsiębiorstwa, to według respondentów zaledwie 17% przedsiębiorstw jest przygotowanych do zaspokojenia aktualnych potrzeb gospodarczych regionu, w 40% model taki winien zostać zmieniony, zaś w 43% model jest już zmieniany (zakłady wchodzące dotychczas w skład kombinatu PGR stają się samodzielnymi przedsiębiorstwami produkcyjnymi).

Z wypowiedzi respondentów wynika, że najwięcej (60%) eskonowało zastosowania informatyki do sterowania procesami technologicznymi, 57% było za automatyzacją śledzenia realizacji zadań planowych, 47% — za usprawnianiem obiegu informacji, 40% — za automatyzacją analiz i kontroli, a 27% — za zastosowaniem komputerów do obliczeń inżynierskich.

We wdrażaniu informatyki respondenci widzieli liczne bariery i ograniczenia. Najwięcej wypowiedzi dotyczyło barier kadrowych (43%) i finansowych (40%). Mniej natomiast organizacyjnych (23%) i psychologicznych (20%), a najmniej technicznych (17%).

Jeśli chodzi o sprzęt do przetwarzania, to 93% przedsiębiorstw dysponuje maszynami do księgowania, 83% — maszynami do fakturowania, 3% — minikomputerami.

1) Jadwiga Orylska: Organizacyjne i ekonomiczne problemy zastosowań informatyki w przedsiębiorstwach rolnych. Problemy zastosowań informatyki w zarządzaniu. TNOiK Oddział w Szczecinie. INFOGRYF'80, Szczecin-Kołobrzeg 1980, s. 149.



Respondenci widzą potrzebę zwiększenia stanu posiadania maszyn księgujących do 100%, maszyn fakturujących — do 90%, maszyn do przygotowania danych na maszynowych nośnikach informacji — do 33% oraz minikomputerów typu MERA 400 — do 17%.

Natomiast opinie respondentów dotyczące lokalizacji przetwarzania były następujące: 57% osób wypowiedziało się za zorganizowaniem rejonowych ośrodków branżowych, obsługujących kilka kombinatów PGR, 33% — preferowało zorganizowanie własnych ośrodków obliczeniowych, a 10% było za przetwarzaniem w ośrodku obcym (np. ZETO).

Na pytanie, jakie systemy należy wdrażać, 40% ankietowanych wypowiedziało się za wykorzystaniem standardowych pakietów programowych, 33% było za wdrażaniem powtarzalnych systemów informatycznych, a 27% — za indywidualnym opracowaniem systemów dostosowanych do potrzeb i warunków konkretnego przedsiębiorstwa.

Prezentowane powyżej opinie kadry kierowniczej PGR należy traktować z dużym przybliżeniem ze względu na małą reprezentatywność grupy (około 30% łącznej liczby PGR-ów regionu). Podane opinie oczywiście nie wyczerpują tematu i mogą stać się przyczynkiem do dalszej dyskusji i pomocą w realizacji zadań reformy.

Uważam, że zagadnienia informatyki powinny znaleźć się w perspektywicznych planach rozwoju działalności PGR jako jedno z istotnych narzędzi usprawniania procesu zarządzania.

W integracji z systemem finansowo-księgowym systemy informatyczne winny zapewnić szeroki serwis informacyjny nie tylko dla celów sprawozdawczych i kontrolnych, lecz również dla potrzeb planowania, z zastosowaniem modeli i metod matematycznych. Pozwoli to na zmniejszenie skutków błędnych decyzji w nowych warunkach.

Dr inż. JADWIGA ORYLSKA  
Akademia Rolnicza w Szczecinie

## Narzędzie ekonomisty?

W Domu Ekonomisty w Warszawie odbyła się 23.09.1982 r. całonocna konferencja pod hasłem: Informatyka narzędziem ekonomisty, zorganizowana przez Komisję Informatyki ZG PTE. W ramach konferencji przedstawiono cztery referaty traktujące o ogólnych problemach zastosowań informatyki oraz cztery — prezentujące konkretne rozwiązania lub koncepcje (np. SIR) informatycznych rozwiązań wspomagających działalność ekonomisty w przedsiębiorstwie. W trakcie dyskusji, nie zawężonej do treści referatów, sformułowano szereg ocen, postulatów i wniosków. Oto najważniejsze z nich:

- Rozpowszechniony dość szeroko pogląd o kryzysie informatyki należy ściśle wiązać ze stanem ekonomicznym kraju, a szczególnie — z funkcjonującym od lat i nadal nie przezwyciężonym brakiem rzeczywistych mechanizmów ekonomicznych regulu-

jących życie gospodarcze. Dopiero bowiem te mechanizmy są w stanie eliminować ozdobne wykwitki informatyczne, służące jedynie prestiżowi czy też czyniące zadość modzie. Mechanizmy te w naturalny sposób wzmacniają sensowne i określają niezbędne zastosowania informatyki.

- Błędem jest tu i ówdzie przyjmowane przez informatyków (choćby w dobrej wierze) zadanie usuwania niedociągnięć organizacyjnych, przepływu dokumentów itp. Działania takie, choć oparte na dobrych chęciach i zapale, powodują nie tylko zniechęcenie, ale przede wszystkim — stanowią alibi dla tych, którzy nie chcą lub nie potrafią usuwać bałaganu leżącego w ich kompetencjach.

- Reczywistość, w jaką — dzięki reformie — ma wejść gospodarka w naszym kraju, zaczyna wymagać coraz więcej informacji. Dotyczy to w szczególności informacji dla przedsiębiorstw — o otoczeniu gospodarczym, o planach i sytuacji państwa. Dla tego celu niezbędne jest wzmocnienie roli, zakresu, jakości i szybkości informacji statystycznej i finansowej. Tej nowej roli aparat GUS i finansów nie jest w stanie spełnić bez istotnego wzmocnienia informatycznego. Jest ze wszech miar słuszne, aby sprzęt komputerowy rozsiany w różnych agendach administracji centralnej skupić na tych dwóch zadaniach. Stworzy to dodatkowo szansę usunięcia tzw. problemu spójności istniejących systemów rządowych.

## GIEŁDA INFORMACJI

### c.d. ITEM

metody dostępu TCAM (z opcją TSO). W niedługiej przyszłości jest przewidziane udostępnienie użytkownikom wersji oferowanego systemu także dla metody BTAM.

Użytkownik systemu ITEM — bezpośrednio na monitorze — interaktywnie tworzy ekranowe mapy w czterech etapach:

- projektowanie postaci znakowej poszczególnych map
- nadawanie atrybutów projektowanej mapie i jej elementom
- podglądanie postaci map dotychczas zaprojektowanych
- powielanie map i ich elementów,

przy czym na każdym z tych etapów można wywoływać moduł HELP do selektywnego wyjaśnienia użytkownikowi sposobów korzystania z systemu, a to znacznie ułatwia wykorzystywanie monitorów ekranowych mało doświadczonym zespołom.

System ITEM zwiększa niezawodność wynikowego oprogramowania użytkowego i stanowi wyjątkowe narzędzie wspomagania projektantów i programistów użytkowych zastosowań konwersacyjnych.

Kontakt merytoryczny: Romuald M. Walewski, tel. 829-13 w 16, Pracownia Zastosowań Numerycznych. Informacje ogólne: dr Ryszard Grzesiak, Dział Handlowy, tel. 829-13 w 72. Adres korespondencyjny: ZETO-Lódź, ul. Narutowicza 136, 90-146 Lódź. Telex: 884-314 ZETO PL.

### Spółka CCL

W sierpniu 1982 r. powstało prywatne przedsiębiorstwo usług informatycznych pod nazwą Spółka Komputerowo-Konsultacyjna CCL. Firma oferuje kompleks usług po cenach konkurencyjnych w stosunku do ofert istniejących placówek państwowych.

Blizsze informacje można uzyskać listownie lub telefonicznie: Spółka Komputerowo-Konsultacyjna — Computer Consulting Leader CCL; mgr mgr Piotr Szempliński, Jacek Zakrzewski, 02-530 Warszawa, ul. Kielecka 28, m. 3; tel. 38-19-35 lub 49-17-47.



• Niezbędne jest zastosowanie narzędzi informatycznych ułatwiających od zaraz trudną sytuację przedsiębiorstw i obywatela, np.:

— usprawnianie reglamentacji (postulowane — ze wskazaniem metod — już od dawna, wielokrotnie)  
— usprawnianie informacji dla przedsiębiorstw poprzez krótkookresowe analizy ekonomiczne państwa metodą szybkich badań reprezentacyjnych  
— organizowanie giełd potrzeb i możliwości (zbyt wiele przykładów świadczy o zmarnowanych szansach, wynikających z braku informacji).

• Kod Towarowo-Materiałowy, który powinien obowiązywać od prawie

dwóch lat, nadal rodzi się w bólach — i to często stając się bardziej hamulcem, niż napędem porządkowania. Niezbędne jest zwiększenie uwagi i aktywności GUS w tym zakresie.

• Istniejącym (i niezbędnym) w przedsiębiorstwach zastosowaniom informatyki zagraża widmo braku części zamiennych, nie mówiąc o braku urządzeń. Z drugiej strony — przedsiębiorstwa produkujące sprzęt komputerowy (np. MERAMAT — produkujący poszukiwane tzw. see-check'i) zamykają produkcję. Bez ingerencji państwa dla powstrzymania tego zjawiska, zastosowaniom informatyki grozi załamanie. Może ono postawić

przyszłość gospodarki kraju bez szans odrobienia strat. Gospodarka 36-milionowego kraju w Europie nie może funkcjonować poprawnie bez szybkiej wymiany informacji.

Komisja Informatyki ZG PTE, zgodnie z wyrażonym na konferencji poglądem zapowiada, iż tego rodzaju spotkania wprowadzone zostaną jako cykliczne imprezy. Wydaje się, że będą one pożytecznym forum dla wymiany poglądów — poszukującym diagnoz i metod działania informatyki w kraju.

J.G.

**ZE ŚWIATA****DATASHOW'82**

Największy japoński pokaz informatyki — salon DATASHOW — po raz dziesiąty odbył się w Tokio w dniach 19—22 października 1982 r. Zorganizowany został przez Japan Electronic Industry Development Association (Japońskie Stowarzyszenie Rozwoju Przemysłu Elektronicznego) i Communication Industries Association of Japan (Japońskie Stowarzyszenie Przemysłu Telekomunikacyjnego). Tegoroczna wystawa imponowała nie tylko wspaniałą organizacją, ale przede wszystkim formą prezentowania sprzętu. Stoiska tętniły życiem: dużo światła, muzyki, projekcji filmów i przezroczy, pokazów laserowych.

Wystawiony sprzęt prawie w 100% był produkcji japońskiej. Pokazano mikrokomputery prezentowane wcześniej w Paryżu (SICOB'82), ale przedstawiono też nowe prototypy maszyn 16-bitowych z wejściami fonicznymi: produkcja firmy NEC rozpoznaje zbiór 120, a TOSHIBA — 256 słów. Wszystkie monitory ekranowe, jakie można było spotkać na wystawie, są już obowiązkowo kolorowe. Kolorowe są też drukowane dokumenty, a przystosowane do tego nowe drukarki będą dostępne na rynku już za kilka miesięcy. Oprócz koloru dominowały również wszelkie formy graficznej prezentacji wyników przetwarzania. Duże zainteresowanie wzbudzały lokalne sieci komputerowe z transmisją światłowodową. Warto przy tym wspomnieć, że japońskie wyroby są na ogół dużo tańsze od swych odpowiedników z firm europejskich lub amerykańskich.

DATASHOW to impreza zorganizowana głównie dla Japończyków (rynek sprzętowy w Japonii jest w 60% oparty przez miejscowych producentów) i ma zadanie ukazania rozwoju technologii japońskiej oraz aktualnych tendencji w informatyce tego kraju. Te tendencje wskazują jednak na to, że już wkrótce cały świat będzie przyjeżdżał do Tokio — i to nie po to, aby wystawiać swoje wyroby, lecz — by kupować tutejsze. Albo przynajmniej — by dowiedzieć się, na jakim etapie naprawdę znajduje się informatyka światowa.

Oprac. **MARIANNA SOBCZYK**  
na podstawie „01 Informatique”

## Informatyka we Francji — kilka liczb

Dane przedstawione poniżej zebrane zostały przez firmę SOREDI, odpowiedzialną za aktualizację bazy danych Centrów Informatyki Francuskiej. Obrazują one stan francuskiej informatyki na dzień 1 lipca 1981 r. Według spisu sprzętu komputerowego Francja posiada 27 655 komputerów (w styczniu 1980 r. było ich 25 391). Oznacza to, że jeden komputer przypada na ok. 1946 mieszkańców tego kraju.

**CHARAKTERYSTYKA PARKU KOMPUTEROWEGO**

Charakterystyka parku komputerowego, sporządzona przez firmę SOREDI, uwzględnia osiem klas sprzętu:

- 0 — komputery biurowe, terminale inteligentne
- 1 — IBM 32 lub DEC PDP 11/20
- 2 — IBM 34 lub DEC PDP 11/34
- 3 — IBM 370/115 lub DEC PDP 11/70
- 4 — IBM 370/138 lub DEC VAX 11/780
- 5 — IBM 370/148
- 6 — IBM 370/158 lub DEC System 10
- 7 — IBM 370/168.

Strukturę francuskiego parku komputerowego, uwzględniającą podaną wyżej klasyfikację oraz udział poszczególnych producentów sprzętu w ogólnej liczbie zainstalowanych komputerów, przedstawia tabela 1.



Tabela 1. Struktura francuskiego parku komputerowego

Firma	Klasa								Razem
	0	1	2	3	4	5	6	7	
IBM	313	1372	3079	1964	1383	700	552	182	9545
CII-HB	—	633	3675	455	1201	360	262	2	5588
OLIVETTI	3572	—	—	—	—	—	—	—	3572
BURROUGHS	2439	138	380	183	189	52	12	3	3396
PHILIPS	2187	27	46	—	—	1	—	—	2261
NCR	—	223	568	693	73	8	1	—	1566
ICL	—	59	188	221	11	7	2	—	488
HEWLETT- -PACKARD	—	—	279	165	—	—	—	—	444
SEMS	—	265	107	40	11	—	—	—	423
UNIVAC	—	50	47	44	26	—	13	29	209
CONTROL DATA	—	—	9	—	80	17	26	17	149
AMDAHL	—	—	—	—	—	—	—	7	7
NASCO	—	—	—	—	—	—	—	7	7
Razem	8511	2767	7378	3765	2974	1145	868	247	27 655

**STOPIEŃ ZINFORMATYZOWANIA POSZCZEGÓLNYCH SEKTORÓW DZIAŁALNOŚCI GOSPODARCZEJ**

Obecność poszczególnych producentów sprzętu komputerowego w najbardziej z informatyzowanych sektorach działalności gospodarczej Francji ilustruje tabela 2.

Tabela 2. Obecność poszczególnych producentów sprzętu komputerowego w najbardziej z informatyzowanych sektorach działalności gospodarczej

Sektor działalności gospodarczej	Liczba zainstalowanych komputerów	Procentowy udział poszczególnych firm <sup>1)</sup>
Handel hurtowy	3599	IBM (36,3), CII-HB (20,8), BURROUGHS (18), NCR (4,7), ICL (1,8), UNIVAC (0,5)
Przemysł dóbr konsumpcyjnych	1988	IBM (44,6), CII-HB (17,4), BURROUGHS (13), NCR (4,6), ICL (2,4), UNIVAC (0,6)
Przemysł dóbr pośrednich	1937	IBM (43,4), CII-HB (19,5), BURROUGHS (12,7), NCR (3,4), ICL (1,9), UNIVAC (1,2)
Badania, doradztwo	1799	IBM (43), CII-HB (29), BURROUGHS (6,9), NCR (3,1), ICL (2,4), UNIVAC (1,2)
Handel — pozostałe	1310	IBM (32,9), CII-HB (21,4), BURROUGHS (13,6), NCR (9,3), ICL (1,6), UNIVAC (0,9)
Instytucje finansowe	1327	IBM (34), NCR (19,7), CII-HB (19,3), BURROUGHS (13,3), ICL (1,6), CONTROL DATA (0,9)
Służby pozahandlowe	1153	IBM (35,6), CII-HB (32,6), BURROUGHS (9), ICL (2,9), NCR (4,9), CONTROL DATA (0,6)
Przemysł rolno-spożywczy	1023	IBM (33,4), CII-HB (22), BURROUGHS (14,9), NCR (4,9), ICL (2,1), CONTROL DATA (0,8)
Administracja ogólna	899	CII-HB (29,7), IBM (28,7), BURROUGHS (9,8), NCR (9,3), ICL (1,3), UNIVAC (0,6)

**MODELE KOMPUTERÓW FIRM MAJĄCYCH NAJWIĘKSZY UDZIAŁ NA RYNKU FRANCUSKIM**

Ostatni spis przeprowadzony przez firmę SOREDI uwidocznił strukturę parku komputerowego poszczególnych firm obecnych na rynku francuskim. W zestawieniu poniższym zobrazowano tę strukturę (pierwszych kilku firm), poczynając od modeli zaliczanych do klasy o numerze najniższym, a skończywszy na modelach zaliczanych do klasy o numerze najwyższym (z wyłączeniem klasy 0) — tabela 3.

Tabela 3

Firma	Modele komputerów w poszczególnych klasach	Liczba instalacji
IBM	S 1, 32, S 7	1372
	34, 3/4 do 3/10; 1130; 38-3	3079
	3/12, 3/15, 38-5; 360/20, 25, 370/115	1964
	360/30, 40; 370/125, 135, 138; 4331	1383
	360/50, 370/145, 148; 4341	700
	3031, 370/155, 158, 360/65	552
CII-HB	3032, 3033, 370/165, 168; 360/67 do 91	182
	Seria 6	633
	G 100, G 10, G 50; Seria 61	2675
	G 150; H 100, 200, 2000/2030; Seria 62, 7720	455
	G 30, G 400; H 2040 do 2100; IRIS 42, 45, 50, 55;	
	10010, 10020; Seria 4004; Seria 64, 7730; Seria 90	1201
BURROUGHS	IRIS 60; 6000 do 6040; Seria 66/05 do 25; 66 DPS	360
	DPS 7, DPS 8; G 600, IRIS 80, 10070; 6050, 6060, 6080; Seria 66/40 do 80	262
	DPS 7/80, 82	2
	Seria L 6000 do L 9000	112
	B 80, 90, 91	26
	B 700, 1700, 1718	380
NCR	B 1721 do 1728; B 1800, B 800, B 500	183
	B 2500/2700/3500/3700	189
	B 4500/4700/4800	52
	B 6700/3800	12
	B 7700	3
	CE 100, 399, 499; 8100	223
ICL	CE 101, CE 151; 8150, 8200, 8230	568
	CE 20J, 8250, 8270; 8410; 8430	693
	CE 300, 8450, 8555; 8550, 8560, 8570	73
	8575, 8580	8
	8585	1
	1300, 1500	59
ICL	S10; 1901, 1902	188
	2903, 2904, 2905; 1903, 1904	221
	1904-S, 2950	11
	2960	7
	2972	2

Opracowała DANUTA SEGIET na podstawie czasopisma LA LETTRE DE L'INDUSTRIE INFORMATIQUE ET DES MARCHES ASSOCIES z dn. 1. 10. 1981 r.

1) Uwzględniono tylko firmy mające udział minimum 0,6%



## O testowaniu, weryfikacji i uwiarygodnieniu programów

Jak nazwać po polsku zespół czynności polegających na sprawdzeniu programów pod względem poprawności i zgodności z wymaganiami, zwanych po angielsku *validation*? Postępowanie to, którego ostatecznym celem jest niejako zaświadczenie o jakości programu, stosuje się coraz częściej przy produkcji oprogramowania. Szerszego upowszechnienia tego podejścia należy oczekiwać wraz z pojawieniem się kompilatorów języka ADA, dla których omawiany proces zostanie zinstytucjonalizowany przez utworzenie tzw. ACVC (ang. *Ada Compiler Validation Center*).

Przed nadaniem polskiej nazwy powyższemu pojęciu, nawet przed wyodrębnieniem najbardziej odpowiednich kandydatów na tę nazwę — warto zastanowić się nad jego znaczeniem. Podamy w tym celu kilka określeń z różnych norm międzynarodowych.

Według normy ANSI (*American National Dictionary of Information Processing*) termin *validation* oznacza sprawdzenie danych pod względem poprawności lub zgodności z odpowiednimi normami, regułami i umowami (*the checking of data for correctness or compliance with applicable standards, rules and conventions*).

Według projektu normy ISO 2382/VIII *validity check* jest to sprawdzenie polegające na upewnieniu się, że dane mieszczą się w ramach określonych ograniczeń, narzuconych przez sprzęt lub oprogramowanie (*the check to ensure that data fall within certain hardware or software limits*). Podobnie norma *Instrument Society of America* określa *validity check* jako sprawdzenie oparte na znajomości ustalonych ograniczeń lub na określonej informacji lub wyniku (*a check based upon known limits or upon given information or results ...*), np. że w roku nie może być więcej niż 12 miesięcy, a tydzień nie może mieć więcej niż 168 godzin.

Zaskakujące jest, że normy nie podają definicji terminu *validation* w odniesieniu do programu. Wszystkie powyższe określenia odnoszą się wyraźnie tylko do danych, nie są więc dla nas zbyt użyteczne.

Spójrzmy wobec tego nieco szerzej na zakres stosowalności omawianego terminu. Okazuje się, że w samej informatyce słowo *validation* występuje w wielu związkach językowych i może oznaczać wiele różnych sytuacji. Tak jak to wynika z norm, może odnosić się do zwykłych zbiorów danych (np. ang. *validation of coefficients, procedure for data validation*), lecz także — do struktur bardziej skomplikowanych (np. *model validation procedure, validating the man-machine interface, validation of a protocol*), może dotyczyć metody (*validation of a heuristic method*), a nawet całego postępowania i jego skutków (*validation of a behaviour simulation*).

Przy tak szerokim zakresie stosowalności terminu może okazać się, że nie jest korzystne dostosowanie jego znaczenia do wszystkich występujących definicji, ani też uogólnienie w tym celu którejs z podanych definicji. Sięgnijmy więc do szczegółowych określeń, które definiują najbardziej interesujące nas pojęcie — *program (software) validation*.

Nie można niestety oprzeć się na normach, a jedynie na propozycjach organizacji normalizacyjnych. Przykładowo — według amerykańskiego *National Bureau of Standards* przez *validation* rozumie się określenie poprawności wytworzonego programu lub oprogramowania, stosownie do potrzeb i wymagań użytkowych. Zwykle dokonuje się

tego przez weryfikację poszczególnych etapów w cyklu rozwojowym oprogramowania (*Validation, Verification and Testing for the Individual Programmer, NBS Special Publication 500-56, February 1980*).

Inna organizacja amerykańska — *American Nuclear Society* podaje następujące określenie pojęcia *program validation*: przetestowanie i ocena pełnego programu komputerowego, mające na celu stwierdzenie zgodności z wymaganiami funkcjonalnymi, jakościowymi i dotyczącymi sprzężeń (ang. *interface requirements*) — *Guidelines for the Verification and Validation of Scientific and Engineering Computer Programs, ANS 10.4 Draft, May 1981*.

W Europie pojęcie to bywa bardzo różnie definiowane, nawet w obrębie jednej organizacji. Na przykład — organizacja EWICS (*European Workshop on Industrial Computer Systems*), mająca ambicje normalizacyjne, podaje następującą definicję: *validation* — uzyskanie pewności, że oprogramowanie odpowiada wymaganiom narzuconym na cały system (*the process of establishing confidence that the software complies with the requirements of the whole system*) — *TC 7 on Reliability, Safety and Security, Working Papers No. 287/295*. Jest to w zasadzie zgodne z definicjami NBS i ANS. Niemniej definicja podana przez Komitet Techniczny nr 11 EWICS (*Glossary of Terms on Application Oriented Specifications*) — proces dostosowywania systemu do narzuconych wymagań (*the process of matching a system with its requirements*) — odnosi się nie tylko do programu, lecz do całego systemu komputerowego.

Zanim jednoznacznie określimy, co oznacza słowo *validation*, warto poszukać związków tego terminu z pojęciami pokrewnymi. Z określeń podanych przez ANS i NBS wynika, że postępowanie zwane *validation* polega na weryfikacji programów na poszczególnych etapach cyklu rozwojowego oprogramowania.

Co w takim razie oznacza weryfikacja programu? Weryfikacja polega na określeniu, czy wyniki każdego etapu w cyklu rozwojowym programu komputerowego spełniają wszystkie wymagania określone na poprzednich etapach (ANS, op. cit.) Dla małych programów, weryfikacja polega na określeniu zgodności programu z jego specyfikacją. W ogólności weryfikacja jest to wykazanie spójności, zupełności i poprawności oprogramowania na każdym etapie i między poszczególnymi etapami cyklu rozwojowego oprogramowania (NBS, op. cit.).

W celu potwierdzenia takiego związku między obydwojoma pojęciami zacytujemy określenie M. S. Deutscha (*Computer, Vol. 14, No. 4, p. 54, 1981*), utworzone na podstawie analizy różnych sposobów używania tych terminów: „Weryfikacja odnosi się do postępowania zapewniającego, że wyniki kolejnego etapu w cyklu rozwojowym oprogramowania odpowiadają intencjom określonym na poprzednim etapie”. Słowo *validation* ma ogólniejsze znaczenie. Celem postępowania jest w tym przypadku upewnienie się, że produkt programowy działa i ma właściwości określone w specyfikacji wymagań (*that a product functions and contains the features prescribed by its requirements specifications*).

Określenia podane w pracach EWICS nie wskazują na taki związek obu pojęć (weryfikacja polega na wykazaniu, że oprogramowanie spełnia określone wymagania — TC 7, op. cit. — lub na udowodnieniu, że system lub program spełnia narzucone wymagania — TC 11, op. cit.).



Podane wyżej określenia organizacji amerykańskich są natomiast zgodne z duchem projektu definicji ISO, która mówi, że: „Weryfikacja jest to czynność polegająca na określeniu, czy operacja została wykonana dokładnie”. Upraszczone sprawę można powiedzieć, formalna weryfikacja polega na dowodzeniu, a empiryczna na testowaniu.

Testowanie natomiast rozumiane jest dość jednoznacznie. Według projektu normy ISO 2382/VIII testowanie jest to wykonywanie programu dla określonego zbioru danych, w celu uzyskania przewidzianego wyniku, będącego podstawą do przyjęcia programu. W podobny sposób definiuje się testowanie w innych dokumentach, np.:

- Testowanie polega na sprawdzeniu zachowania się programu przez wykonanie go dla próbnych zbiorów danych (NBS, op. cit.).
- Testowanie polega na wykonaniu oprogramowania w celu określenia, czy otrzymane wyniki są poprawne (EWICS TC 7, op. cit.).
- Testowaniem nazywa się wykonanie programu (w dokładnym brzmieniu normy — systemu) w ściśle określonych warunkach i porównanie jego zachowania z oczekiwanym, w celu wykrycia i umiejscowienia błędów i uszkodzeń (EWICS TC 11, op. cit.).

Wróćmy teraz do problemu sformułowanego na początku: jak nazwać po polsku *validation*?

Wszystkie podane definicje terminu *program validation* łączy występowanie słowa *wymaganie*. Zatem podczas procesu określanego mianem *validation* chodzi o wykazanie, że program (oprogramowanie) spełnia określone wymagania, jest zgodny z wymaganiami lub odpowiada określonym wymaganiom itp. Wszystkie te sformułowania oznaczają w praktyce to samo.

Mniej ważny jest rodzaj tych wymagań, gdyż zależą one od rozwiązywanego problemu — a więc tej cechy nie można wziąć pod uwagę i uwzględnić w definicji. Bardziej istotny jest fakt upewnienia się, że te wymagania są rzeczywiście spełnione. W praktyce zatem postępowanie określane mianem *validation* polega na uczynieniu programu przedmiotem tej pewności.

Według „Słownika języka polskiego” (Pod red. W. Doroszewskiego, PWN, Warszawa 1981) *przeświadczenie*, czy

pewność, że coś jest prawdziwe, nazywa się *wiarą*. Jeżeli zatem mowa o danych, to zadanie polega na uzyskaniu *przeświadczenia*, pewności, że dane są prawdziwe, a więc na tym, aby uczynić je godnymi wiary, wiarygodnymi, tj. aby je uwiarygodnić.

Jeżeli powyższe rozumowanie rozszerzymy na program, to można powiedzieć, że proces zwany *validation* polega na uczynieniu programu przedmiotem pewności, że jest zgodny z wymaganiami, a więc — na uwiarygodnieniu go.

Doszliśmy zatem do stwierdzenia, że *validation* to po polsku uwiarygodnienie. Choć proponowana nazwa jest nieco kontrowersyjna, wiernie oddaje treść pojęcia. Ma ponadto tę zaletę, że jest elastyczna i trafnie wyraża istotę różnych związków językowych, np. wymienionych na początku artykułu: wiarygodne współczynniki (ang. *valid*), uwiarygodniony protokół (ang. *validated*), wiarygodność modelu (ang. *validity*), uwiarygodnienie metody (ang. *validating*). Trudno byłoby znaleźć drugą nazwę o podobnych właściwościach językowych.

Proces uwiarygodnienia może prowadzić do formalnego uznania programu ze względu na spełnienie określonych wymagań lub pod względem przydatności do określonego celu. Czynność ta nazywana po angielsku *certification*, ma cechy legalizacji (ang. *to certify* — zaświadczać na piśmie). Proponowałbym więc określić ją jako zatwierdzenie programu lub uprawomocnienie programu.

Według wymienionej pracy NBS, *certification* jest to przyjęcie oprogramowania przez upoważnioną osobę, po uprzednim uwiarygodnieniu go przez tę osobę lub po wykazaniu tej osobie jego wiarygodności.

Pojęcie to wystąpiło także w literaturze polskojęzycznej. W przetłumaczonej na język polski książce G. W. Myersa — „Projektowanie niezawodnego oprogramowania” (WNT, Warszawa, 1980, tłum. St. Matwin) nazwano je *certyfikacja* (str. 164). W wymienionej książce wystąpiły także pozostałe omawiane terminy, jednak treść odpowiednich definicji nasuwa znaczne wątpliwości, dlatego nie będę ich cytował odsyłając zainteresowanych Czytelników do książki. Warto jednak zaznaczyć, że słowo *validation* zastąpiono w tłumaczeniu słowem *konfirmasiacja*.

JANUSZ ZALEWSKI

## Bibliografia wydawnictw polskich z dziedziny informatyki

• Materiały pomocnicze do laboratorium ETO. Język FORTRAN maszyn serii ODRA 1300 — GUZIAK T. (kier. pr. zb.), Wyd. Politechniki Lubelskiej, Lublin 1981, s. 310, cena 30 zł

Wyrażenia arytmetyczne. Segment MASTER. Instrukcja podstawienia arytmetycznego. Instrukcja we/wy. U uruchamianie programów w systemie BATCH MONITOR FORTRAN przy użyciu kompilatora XFAS. Schematy blokowe. Instrukcje sterujące w FORTRANIE. Instrukcja DO. Tablice i zmienne indeksowe. Wyrażenia logiczne i ich wykorzystanie. Podprogramy. Funkcje niestandardowe określone wyrażeniem. Segment FUNCTION. Segment SUBROUTINE. Wywołanie segmentu FUNCTION i SUBROUTINE. Deklaracja EXTERNAL. Deklaracje COMMON i EQUIVALENCE. Przetwarzanie danych nienumerycznych. Wykorzystanie podprogramów bibliotecznych.

Skrypt zawiera podstawowe informacje o konstruowaniu programów w języku FORTRAN. Przeznaczony jest dla studentów uczelni technicznych specjalizujących się w dziedzinie informatyki.

• Komputerowy system przetwarzania tekstów fonematycznych — BOLC L., MAKSYMENKO M., Wyd. Uniwersytetu Warszawskiego, Warszawa 1981, s. 125, cena 23 zł

Automatyczna translacja fonematyczna tekstów ortograficznych. Analiza statystyczna tekstów fonematycznych. Analiza algorytmów. Opis programów. Opis danych. Omówienie wyników. Materiały przeznaczone są dla pracowników naukowych zajmujących się automatyczną translacją fonematyczną tekstów.

• Układy mikroprocesorowe serii INTEL 8080, MOTOROLA 6800, Am 2900 — PAWŁOWSKI M. i inni. (pr. zb.), Wyd. Politechniki Warszawskiej, Warszawa 1981, s. 228, cena 20 zł

Rodzina układów MCS 80 firmy INTEL, M 6800, Am 2900. Skrypt przeznaczony jest głównie dla studentów Wydziału Elektroniki kierunku Informatyka jako materiał pomocniczy dla laboratorium mikroprocesorów.



• Organizacja maszyn cyfrowych w ujęciu strukturalnym, TAN-NENBAUM A. S. Tłum. wyd. ang. z 1976 r. WNT, Warszawa 1980, s. 517, cena 120 zł

Organizacja maszyn cyfrowych. Konwencjonalny poziom maszynowy. Poziom mikroprogramowania. Poziom systemu operacyjnego maszyn. Poziom języka assemblera. Maszyny wielopozimowe. Sygnały dotyczące dalszej lektury i wykaz literatury. Dodatki: Arytmetyka o skończonej precyzji i liczby dwójkowe. Liczby zmiennie-pozycyjne. Algebra Boolea.

Książka ma służyć jako podręcznik do wstępnego wykładu z programowania w języku assemblera i organizacji maszyn cyfrowych. Książka jest zrozumiała dla studentów informatyki pierwszych lat studiów. Może być przydatna programistom i użytkownikom maszyn cyfrowych.

• Od algebry połączonych do mikroprocesora — PELWAM. Tłum. wyd. niem. z 1977 r. WKiŁ, Warszawa 1980, s. 236, cena 38 zł

Kody. Układy logiczne. Przerzutniki. Monowibrator. Liczniki. Układy czasowe. Pomiar czasu. Pomiar częstotliwości. Pomiar przebiegów periodycznych. Układy arytmetyczne. Technologie i rodzaje układów scalonych. Cyfrowe układy sterowania. Mikroprocesor — mikrokomputer.

Książka przeznaczona jest dla szerokiego kręgu czytelników zainteresowanych techniką cyfrową.

• Elektroniczna technika obliczeniowa — SAPIECHA K. WNT, Warszawa 1981 r., s. 283, cena 37 zł

Pojęcie algorytmu i jego reprezentacja. Zasady działania elektronicznych systemów liczących. Podstawy programowania elektronicznych systemów liczących. Programowanie w języku FORTRAN 1900. Podstawy użytkowania systemów liczących. Dodatki.

Podręcznik przeznaczony jest dla studentów wydziału elektroniki wyższych uczelni technicznych.

• Podstawy projektowania wielkich sieci teleinformatycznych. Tłum. wyd. z ros. z 1976 r. — ŻYMIERIN D. G., MAKSYMIEŃKO W. I. — red. WNT, Warszawa 1981, s. 294, cena 80 zł

Podstawowe sposoby projektowania sieci teleinformatycznych. Wielodostępne ośrodki obliczeniowe (WOO) na bazie Jednolitego Systemu Elektronicznych Maszyn Cyfrowych (JSEMC). Transmisja danych w sieciach ośrodków obliczeniowych. Określenie podstawowych charakterystyk technicznych i lokalizacji sieci ośrodków obliczeniowych. Wyposażenie informatyczne. Oprogramowanie sieci ośrodków obliczeniowych. Zadania rozwiązywane przez sieć ośrodków obliczeniowych.

Książka przeznaczona jest dla projektantów i użytkowników systemów informatycznych.

• Projektowanie struktur systemów operacyjnych — OLSZEWSKI J. WNT, Warszawa 1981, s. 420, cena 27 zł

Programy, procesy i ich struktury. Wybór przykładów. Konstruowanie systemu operacyjnego.

Dodatek: składnia ukierunkowanej maszynowo wersji Pascala. Książka przeznaczona jest dla programistów.

• Zbiór zadań z informatyki — BAUER F., GNATZ R., HILL U. Tłum. wyd. niem. z 1975 r. i 1976 r. WNT, Warszawa 1981, s. 349, cena 70 zł

Informacja i wiadomości. Podstawy programowania. Języki algorytmiczne zorientowane maszynowo. Sieci przelączające i sekwencyjne. Organizacja danych i procesów. Automaty i języki formalne. Semantyka języków algorytmicznych. Dodatek: Metodyka programowania.

Książka przeznaczona jest dla studentów informatyki oraz programistów.

• Komputerowe wspomaganie systemu sterowania jakością — JANISZEWSKI J., MOSZKOWSKA-KOWALEW E. Wyd. Towarzystwa Naukowego Organizacji i Kierowania — Oddział w Bydgoszczy, Bydgoszcz 1981, s. 133

Biblioteka poradników jakości. Materiały szkoleniowe. Ogólne zadania jednostek organizacyjnych wszystkich szczebli zarządzania w zakresie zapewnienia jakości, wynikające z obowiązujących aktów normatywnych. Zasady tworzenia informacji dla celów sterowania i zarządzania jakością (system informacyjny). Komputerowe wspomaganie systemu sterowania jakością. Ogólne zasady wdrażania systemów zapewnienia jakości ze wspomaganie komputerowym. Nakłady i efekty. Materiały przeznaczona są dla kadry kierowniczej przedsiębiorstw produkcyjnych.

• Organizacja przesyłania informacji w systemach cyfrowych — DUDEK Z. T., SOSNOWSKI J. PWN, Warszawa 1981, s. 289, cena 40 zł

Cz. 1. Zasady przesyłania informacji w systemie cyfrowym; zagadnienie przesyłania informacji. Struktury sieci przesyłania informacji. Technika przesyłania informacji. Zasady przesyłania informacji w łączach. Dopasowanie trybu pracy nadajnika do trybu pracy odbiornika. Sterowanie przepływem informacji w systemie cyfrowym.

Cz. 2. Zasady wprowadzania i wyprowadzania informacji w systemach cyfrowych; klasyfikacja urządzeń wejścia — wyjścia. Urządzenia wprowadzania i wyprowadzania informacji w postaci cyfrowej. Wyprowadzanie informacji w postaci graficznej. Wprowadzanie informacji w postaci graficznej.

Cz. 3. Zasady organizacji współpracy urządzeń zewnętrznych z systemem komputerowym; układy wejścia-wyjścia maszyn cyfrowych. Programowanie przesyłania danych. Autonomiczne przesyłanie danych.

Książka przeznaczona jest przede wszystkim dla studentów specjalizujących się w cyfrowych systemach informatyki i automatyki oraz dla inżynierów projektantów tych systemów.

• Wprowadzenie do metodologii projektowania banków danych technologicznych — GONTARCZYK T. PWN, Warszawa — Łódź 1981, s. 90, cena 30 zł

Rozwój i zastosowanie metodologii projektowania systemów informacyjnych. Sformułowanie zadania, problemu w zakresie tworzenia banku danych technologicznych. Studium wstępne budowy danych technologicznych. Projektowanie banku danych technologicznych. Opracowanie szczegółów projektu. Oprogramowanie i wdrożenie banku danych technologicznych.

Materiały przeznaczone są dla pracowników przedsiębiorstw przemysłowych, zwłaszcza technologów, zainteresowanych problematyką komputerowo wspomaganym metod projektowania.

• Ekonomiczne problemy zautomatyzowanych systemów zarządzania — KISIELNICKI J. PWE, Warszawa 1981, s. 257, cena 50 zł „Informatyka w praktyce”

Systemy informacyjne w jednostkach gospodarczych. Kierunki zastosowań zautomatyzowanych systemów zarządzania. Efektywność zastosowań i czynniki na nią wpływające. Ocena efektywności zautomatyzowanych systemów zarządzania za pomocą rachunku ekonomicznego. Elementy rachunku ekonomicznego i wybrane metody ich określania.

Książka przeznaczona jest dla projektantów i użytkowników zautomatyzowanych systemów zarządzania.

• Metody informatyczne — KISIELNICKI J. PWE, Warszawa 1981, s. 294, cena 40 zł

Badania i analiza systemów informacyjnych. Projektowanie danych wejściowych i kontrola. Projektowanie danych wyjściowych. Projektowanie elementów systemów informatycznych. Ulepszone metody tworzenia oprogramowania. Projektowanie całościowych systemów informatycznych. Projektowanie systemów informatycznych opartych na bazie danych. Ochrona danych w systemach informatycznych. Ocena systemów informatycznych. Książka przeznaczona jest dla działaczy gospodarczych, projektantów informatycznych systemów zarządzania i studentów wyższych uczelni ekonomicznych.

• Funkcje administratora bazy danych i administratora zastosowań — BRANDT A. Wyd. Zjednoczenie Informatyki. Centrum Projektowania i Zastosowań Informatyki, Warszawa 1981, s. 124, cena 82 zł

„Problemy Informatyki”

Specyfika projektowania bazy danych. Administrator bazy danych. Administrator zastosowań. Obszary bezpośredniej współpracy administratora bazy danych i administratora zastosowań. Kierunki rozwoju funkcji administracyjnych w otoczeniu bazy danych. Materiały przeznaczone są dla projektantów systemów informatycznych i kadry kierowniczej ośrodków obliczeniowych.

• Wybrane metody oceny prac projektowych i programowych. Praca zbiorowa. Wyd. Zjednoczenia Informatyki. Centrum Projektowania i Zastosowań Informatyki, Warszawa 1981, s. 124, cena 82 zł

„Problemy Informatyki”

Metoda oceny jakości prac projektowych. Wykorzystanie techniki PSLPSA do oceny prac projektowych. Metody oceny prac programowych. Zalety i wady przedstawionych metod oceny oraz ich przydatność w warunkach polskich. Materiały przeznaczone są dla kadry kierowniczej ośrodków obliczeniowych.

Oprac. A.K.



## Selekcja negatywna

Trend, o którym piszę jest bardzo niepokojący, szczególnie, gdy przyjąć dłuższą perspektywę. Negatywna selekcja ludzi zajmujących się informatyką ma miejsce nadal, mimo że zapowiadana reforma miała ją zlikwidować — zmienić na właściwą.

Podzielmy „informatyków” na działaczy w informatyce i zawodowców. Taki podział ułatwi dalszy wywód.

Do działaczy informatyka w Polsce nie miała szczęścia. Pierwsi z nich, nie mając dostatecznego doświadczenia i wiedzy, wypracowali opóźnienie 6—8 lat w stosunku do czołówki (start nie był wprawdzie wyrównany, ale w biegu nie wyszliśmy z dołków ostatni, niegdyś byliśmy nawet awangardą KDL'ów). Druga generacja działaczy, zafascynowana wizją Polski — potęgi gospodarczej, „oderwała się od mas”. Siermiężna rzeczywistość informatyczna walczyła tymczasem z brakiem papieru, taśm, złymi konfiguracjami, zawodnością urządzeń, niezrozumieniem ani roli, ani możliwości komputerów i dziesiątkami innych barier.

Charakterystyczne, że wówczas — mimo owej budującej wizji — niewiele uczyniono dla tak decydującej sprawy, jak kształcenie — kształcenie porządne, w wyposażonych w komputery uczelniach. Wizje miały jednak to do siebie, że informatyka nabrała rozgłosu; być informatykiem — to było to. Pasowano ich szybko i masowo, po kursach lub pogadankach telewizyjnych. Rosły nowe firmy, nowe ośrodki, przybywali nowi działacze. Stałym postulatem środowiska, rzadko zresztą spełnianym, było awansowanie działaczy z samej branży, nie zaś asymilowanie ludzi sprawdzonych gdzie indziej.

Boom trwał jednak krótko, w oczach wysokich sfer informatyka traciła swój blask, zaczęła niekiedy budzić wątpliwości. Trudno się więc dziwić, że ludzi dobierano według zasady: sprawdzony, a nie — zna się na rzeczy. (Wyjątki nie wpłynęły istotnie na ogólny obraz).

Przygaszenie blasku informatyki związane na dodatek z poglądem władzy o nieokreślonej efektywności informatyki (czytaj: kapitałozerna, a bez efektów) zahamowało rozwój w chwili, kiedy dopiero coś zaczęło się dziać, zaczęli opuszczać mury uczelni dobrze wykształceni informatycy, tu i ówdzie pojawiły się udane zastosowania, wytrenowane ekipy radziły już sobie z zawodnością, rysowały się perspektywy (niskiej, bo niskiej) stabilizacji.

Na koniec przyszedł otwarty kryzys — środowisko informatyków nie było specjalnie zdumione, miało nawet nadzieję, że wreszcie uda się namówić kogo trzeba na racjonalne działanie, na zdjęcie hamulców, że ostra selekcja zmieni być może znak na dodatni.

Rok 1980 to początek: rozpoczął się odpływ ludzi z informatyki, niestety głównie informatyków i to tych ruchliwych, niecierpliwych, gnanych potrzebą sukcesu, nie mogących znieść atmosfery zębności i nieukrywanej podejrzliwości. A pamiętajmy — na świecie zapotrzebowanie na informatyków gwałtownie wzrosło, ekspansja mikroprocesorów żąda progamistów. W żadnym kraju na listach bezrobotnych nie spotyka się informatyków, zatrudnia się ich nawet nielegalnie, a i tak ciągle występują niedobory.

W Polsce zaś zaczyna się mówić o bezrobociu informatyków. W okresie ostatniego roku zmalało zatrudnienie — szacuje się, że o ok. 20%. Gdyby zmalała tzw. masa tabulek to pół biedy, odchodzą jednak fachowcy, niekiedy — muszą odejść. Niestety rzadko odchodzą działacze, a jeśli — już, to też niekoniecznie z merytorycznych powodów.

Wprawdzie ludzi zatrudnionych w informatyce jest jeszcze w Polsce wielu, niemniej ich liczba maleje. Maleje też prestiż (choćby: średnia płaca wynosi ok. 2/3 średniej krajowej). W efekcie coraz mniej jest chętnych do zawodu, coraz słabszy przepływ wiedzy, myśli, poglądów.

A wystarczyłoby tylko jedno, na początek: zmienić znak selekcji.

JANUSZ GWIAZDA

## Ot, marzenie

Selekcja pozytywna? Decyzje podejmowane przez fachowców, potrafiących realnie ocenić możliwości i zagrożenia informatyki? Awansowanie „zawodowców” zamiast „działaczy w informatyce”? Kształtowanie menażerów, nie zaś narzucanie ich według odgórnego rozdzielnika? Stosowanie konkursów, by na każde stanowisko wybrać najlepszego? Ależ naturalnie — chcieliśmy, chcemy, będziemy chcieli nadal.

Ale — mimo nawet, bywało, głośniego protestu — nadal awansuje się „działaczy”. Ci zaś — którzy wszystko zawsze wiedzą najlepiej i którzy w imię reformy chętnie nałożyliby gospodarce sztywny kaftan, by poruszała się potulnie, na rozkaz — wierzą zapewne, że prawa ekonomiczne można opinać, a o wszystkim zdoła rozstrzygnąć dyrekturywa. Zadufani w sobie nie biorą jednak pełnej odpowiedzialności za swe decyzje, przerzucając ją w efekcie na barki fachowców. Można optymistycznie przyjąć, że popełniane błędy przestaną być wreszcie u nas bezkarne i że ktoś będzie musiał za nie odpowiedzieć. Warto przewidzieć to zawczasu.

Postulat selekcji pozytywnej jest w istocie propozycją dla władz centralnych, by uznały środowisko informatyków za podmiot ich własnej pracy, za źródło koncepcji — związanych zarówno z podstawową technologią, jak i me-

todami organizacji informatyki oraz sposobem wykorzystania tego tworzonego przez siebie narzędzia. Niższe szczeble hierarchii państwowej (w tym — informatyków) traktuje się jednak jako wykonawców, nie zaś — kreatorów rzeczywistości. A że „działacze” są z reguły kiepskimi fachowcami, to wynika niejako z natury rzeczy i choćby, że się poprawią bądź że ustąpią miejsca lepszym, to i tak — łączy co najmniej naiwnością.

Informatykom nie pozostaje zatem chyba nic innego, jak pogodzić się z prymatem ludzi niekompetentnych (ale dostatecznie „wiarygodnych”) i pod ich rządami — robić swoje.

Ważne są fakty: konkretne opłacalne realizacje, dynamiczna działalność małych zespołów fachowców. O te fakty nie jest wprawdzie łatwo, ale też nie są one niemożliwe. Wyraźnie bowiem odstąpiono od wykorzystywania informatyki dla wzmocnienia struktur państwowych; cele społeczne coraz wyraźniej dochodzą do głosu. Ekonomia zaś silnie wspiera ten „dezintegracyjny” proces.

Myszę, że selekcję pozytywną wymuszą z czasem konieczności ekonomiczne (których skuteczność można zwiększyć własnym działaniem). Wszystkie zaś apele, jeśli przyniosą, to tylko odwrotny skutek.

ZBIGNIEW GLUZA



# robot

BIBLIOTEKA GŁÓWNA  
Politechniki Śląskiej

1877 / 82

Wszędzie poszukiwani są specjaliści, zwłaszcza o dużych walorach adaptacyjnych. Nasz system A 5120 spełnia wszystko to, co przyrzekliśmy. Dostawiamy wszędzie: w handlu, przemyśle, ośrodkach obliczeniowych, komunikacji i instytucjach finansowych.

A 5120 stosowany być może również w trybie zdalnym jako terminal. Składa się on z monitora oraz jednostki pamięci na dysku elastycznym lub jednostki pamięci kasetowej. Jego wydajność znacznie wzrasta przez wyposażenie w drukarkę oraz dodatkowe jednostki pamięci na dysku elastycznym lub pamięci kasetowej.

Rejestrowane dane można wprowadzać na zewnętrzne nośniki informacji lub bezpośrednio do jednostki centralnej. I jeszcze jedna istotna cecha: dzięki przyjęciu koncepcji pół ekranu A 5120 jest szczególnie dobrze przystosowany do specyfiki rejestracji danych masowych.

Użytkownik ma do dyspozycji gotowe moduły programów oraz kompletne systemy programowe.

Wszechstronny  
specjalista  
wkracza do biur  
Nowy przedstawiciel  
rodziny A 5100

Na Wiosennych Targach  
Lipskich

od 13 do 19.03.1983 r.

wystawa w hali 15

Biurowe Handlowe

w hali 12.12

Prosimy zwrócić się o szczegółowe informacje oraz dokładne obejrzenie A 5120, a następnie przygotować miejsce dla zainstalowania tego systemu księgująco-obrachunkowego.

**robotron**

Robotron Export-Import  
Państwowe Przedsiębiorstwo  
Handlu Zagranicznego  
Niemieckiej Republiki  
Demokratycznej  
DDR 1080 Berlin,  
Friedrichstrasse 61

## A 5120

