POLITECHNIKA ŚLĄSKA
BIBLIOTEKA GŁÓWNA
P. 1877/75

4 1975

informatyka

SPIS TREŚCI NR 4/75

		Str.
	O stanie i kierunkach rozwoju teorii systemów przełączających *Tadeusz Puchałka*	1
	Przegląd wybranych metod wykorzystania i organizacji pamięci systemu cyfrowego *Mieczysław Muraszkiewicz*	6
	Organizacja systemów wielokomputerowych w oparciu o komputery ODRA 1305 *Andrzej Dzierżykraj, Wojciech Kuczborski*	12
	Niepewna przyszłość francuskiego przemysłu informatyki *Jerzy Miłek*	16
	Komputeryzacja prac edytorsko-poligraficznych. Fotoskład *Jędrzej Wróblewski*	18
	Struktury nakładkowe programów w systemie DOS/JS *Lech Jagodziński*	24
	Dialog Sokratyczny — tłum. i oprac. *Stanisław Matwin*	29
	Formy organizacyjne pionów informatyki — tłum. i oprac. *Ludwik J. Rossowski*	31
INFORMATYKA NA UCZELNIACH	Informatyka na Uniwersytecie Edynburskim — *Gerard Zieliński*	32
Z KRAJU	Problemy wielodostępnych systemów informatycznych — *Zenon Grodzki, Andrzej Piaskowski*	34
	Informatyka w przemyśle siarkowym	35
ZE ŚWIATA	Informacje różne	36—37
OŚRODKI INFORMATYKI PREZENTUJĄ	Informatyka w centrali handlu zagranicznego — *Barbara Nowakowska*	38
TERMINOLOGIA	Programowanie komputerów cyfrowych. Terminy ogólne — *Władysław Klepacz*	40
TRYBUNA CZYTELNIKA	*List Ludwika J. Rossowskiego*	41
PROJEKTOWANIE SYSTEMÓW EPD	Metody analizy kompleksowej systemu informacyjnego przedsiębiorstwa — *Aleksander Adamowicz, Stanisław Szklarzewicz*	42
PROBLEMATYKA BAZY DANYCH	Metody oceny efektywności systemów zarządzania bazą danych — *Józef Czajkowski*	45
PRZEGLĄD WYDAWNICTW	Bibliografia wydawnictw polskich z dziedziny informatyki — oprac. *Jerzy Klamborowski*	48, 49 i IV okł.

WYDAWNICTWA CZASOPISM TECHNICZNYCH NOT

Warszawa
Czackiego 3/5

KOLEGIUM REDAKCYJNE

Redaktor Naczelny prof. dr hab. Leon ŁUKASZEWICZ

mgr Krystyn BERNATOWICZ, prof. dr hab. inż. Konrad FIAŁKOWSKI (zast. redaktora naczelnego), doc. dr inż. Zbigniew GACKOWSKI, mgr inż. Marek HOŁYŃSKI, Władysław KLEPACZ, Elżbieta KOŁODZIEJSKA, doc. dr hab. Antoni MAZURKIEWICZ,

Sekretarz Redakcji mgr Krystyna WROŃSKA — Red. tech. Józef DUSZA

RADA PROGRAMOWA

Mgr inż. Antoni BOSSOWSKI, doc. dr inż. Jan FELICKI, doc. dr inż. Zbigniew GACKOWSKI, doc. dr inż. Aleksander GOLINOWSKI, dr hab. Andrzej GRZYWAK, doc dr hab. inż. Roman KULESZA, prof. dr hab. Leon ŁUKASZEWICZ, prof. dr hab. inż. Stanisław PASZKOWSKI (wiceprzewodniczący), prof. dr Tadeusz PECHE, mgr inż. Bronisław PIWOWAR, dr inż. Andrzej PLĄSKOWSKI, mgr inż. Tadeusz PODGÓRSKI (wiceprzewodniczący), prof. dr inż. Jerzy SEIDLER, prof. dr inż. Andrzej STRASZAK (przewodniczący), doc. Jerzy TRYBULSKI, dr Tadeusz WALCZAK, prof. dr inż. Stefan WĘGRZYN, dr inż. Jan Z. ŻYDOWO

Redakcja: 00-041 Warszawa, ul. Jasna 14/16, pokój 331, tel. 27-71-40 lub centrala 28-82-61 w. 285, dyżury redakcji 10,00—13,00

Zakład Kolportażu WCT NOT, Warszawa, ul. Mazowiecka 12

Zakł. Graf. „Tamka". Z. 2. Zam. 118. Papier druk. sat. IV kl. 70 g 61 + 86. Obj. 6,75 ark. druk. Nakład 5600 + 30. B-85.

Cena egzemplarza zł 15.— INDEKS 36210/36124 Prenumerata roczna zł 180.—

Informatyka

dawniej Maszyny Matematyczne

zastosowania w gospodarce, technice i nauce

Nr 4

MIESIĘCZNIK

1975

ROK XI

Kwiecień

ORGAN KRAJOWEGO BIURA INFORMATYKI I POLSKIEGO KOMITETU AUTOMATYCZNEGO PRZETWARZANIA INFORMACJI NACZELNEJ ORGANIZACJI TECHNICZNEJ

P. 1877/75

TADEUSZ PUCHAŁKA
Politechnika Poznańska

681.325.65

O stanie i kierunkach rozwoju teorii systemów przełączających

Artykuł przedstawia ewolucję teorii systemów przełączających, koncepcję klasyfikacji tych systemów oraz opis algebraiczny ich poszczególnych klas. Omówiono również problematykę reprezentacji i rozpoznawania języków przez automaty oraz możliwości techniczne zastosowań modeli systemów przełączających.

1. Ewolucja teorii systemów przełączających

Teoria systemów przełączających jest młodą dyscypliną naukową. Jej początki sięgają pierwszych lat drugiej połowy naszego stulecia. W tym czasie zaczynają się formować podstawy teoretyczne układów kombinacyjnych. Uwaga badaczy skupia się na układach przekaźnikowo — stykowych, stanowiących obszerną klasę układów kombinacyjnych. Przedmiotem szczególnego zainteresowania są zagadnienia syntezy tych układów. W 1950 r. ukazuje się praca M. A. Gawriłowa [11], poświęcona problematyce teorii układów przekaźnikowo — stykowych.

Układy kombinacyjne są systemami przełączającymi bez możliwości zapamiętywania sygnałów. Przyjęcie postulatu, iż każdy stan systemu przełączającego w chwili $t+1$ zależy od stanu systemu i sygnału wejściowego w chwili t, w jakościowy sposób zmienia jego meta-strukturę. Przełom lat pięćdziesiątych i sześćdziesiątych wiąże się z koncentracją badań nad zdeterminowanymi systemami przełączającymi o synchronicznej organizacji czasu i o określonej wyżej strukturze przejść od stanu do stanu. Systemy te nazwano zdeterminowanymi automatami synchronicznymi. W 1959 r. M. O. Rabin i D. Scott w pracy [33] formułują podstawy pojęciowo-definicyjne automatów skończonych, tj. takich, których moc zbioru stanów jest skończona. Podejmowane są z powodzeniem próby stworzenia głębokiej, algebraicznej teorii automatów zdeterminowanych [15], umożliwiającej wnikliwe badanie własności oraz dokonywanie ich syntezy na poziomach abstracyjnym i strukturalnym.

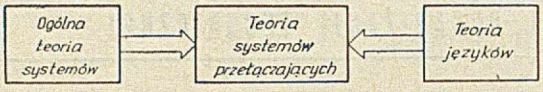
Prof. dr TADEUSZ PUCHAŁKA (ur. 1929) ukończył Wydział Elektromechaniczny Akademii Górniczo-Hutniczej w Krakowie. Jest profesorem nadzwyczajnym w Politechnice Poznańskiej, gdzie kieruje Zakładem Automatyki Przemysłowej. Prowadzi badania naukowe w dziedzinie teorii i zastosowań systemów sterowania oraz teorii systemów przełączających (70 publikacji). Współpracuje z przemysłem wielkopolskim w dziedzinie systemów automatyzacji kompleksowej. Członek międzynarodowych i krajowych towarzystw i organizacji naukowych.

W automatach zdeterminowanych przejście od stanu do stanu pod działaniem symbolu wejściowego, dokonuje się z prawdopodobieństwem $p = 1$. Uogólniając meta-strukturę automatu można założyć, iż prawdopodobieństwo przejścia od stanu s_i do stanu s_j, pod działaniem symbolu wejściowego x, zawarte jest w przedziale [0,1] ($p(s_i/s_j, x) \in [1]$). Automaty o takiej meta-strukturze nazwano automatami stochastycznymi (probabilistycznymi). M. O. Rabin w 1963 r. ogłasza pracę [32], w której zawiera podstawy pojęciowo-definicyjne tej klasy automatów.

Zgodnie z ogólną tendencją ostatnie lata przynoszą wzrost zainteresowania badaczy jeszcze bardziej złożonymi strukturami systemów przełączających. Przedmiotem szczególnie intensywnych badań są wolnostukturowe automaty dendrytowe typu opadającego i wznoszącego, automaty grafowe oraz inne automaty złożone. W 1970 r. M. Magidor i G. Moran publikują pracę [24], która zawiera podstawy pojęciowo-definicyjne stochastycznych wolnostrukturowych automatów dendrytowych oraz dokonują badania rozpoznawania (akceptowalności) języków bezkontekstowych przez te automaty. W omawianych klasach automatów zbiory stanów oraz funkcje przejść skonstruowane są bardzo ogólnie.

2. Klasyfikacja systemów przełączających. Zakres problemowy

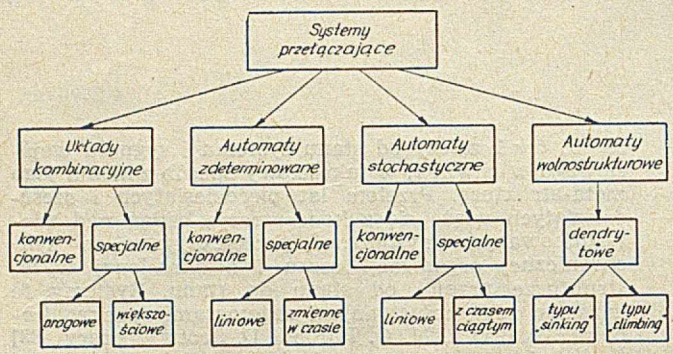
Teoria systemów przełączających jest teorią komplementarną, obejmującą swym zakresem zarówno problemy systemowe jak i językowe oraz ich ścisłe związki. Stąd też tak istotne znaczenie posiadają badania, dotyczące akceptowalności jak i reprezentowalności języków przez automaty, szczególnie w zakresie automatów stochastycznych i o złożonych strukturach. Wzajemne związki, jakie istnieją pomiędzy ogólną teorią systemów i teorią języków, w relacji do klasy systemów przełączających, przedstawiono na rysunku 2.1.



Rys 2.1

Rys. 2.1 Schematyczne przedstawienie związków teorii systemów przełączających z ogólną teorią systemów i teorią języków

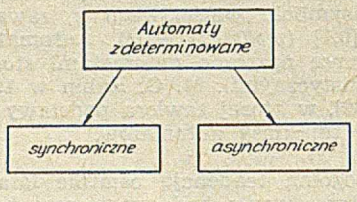
W dalszych rozważaniach przyjęto konwencję klasyfikacji systemów przełączających wg ich meta-struktury. Klasyfikacja zobrazowana jest na rysunku 2.2.



Rys. 2.2.

Rys. 2.2 Klasyfikacja systemów przełączających według ich meta-struktury

Z punktu widzenia organizacji czasu, możemy w klasie automatów zdeterminowanych wyróżnić dwie podklasy: automaty synchroniczne i automaty asynchroniczne, co zilustrowano na rysunku 2.3.



Rys. 2.3

Rys. 2.3 Klasyfikacja automatów zdeterminowanych według sposobu organizacji czasu

Główny obszar problemowy teorii układów kombinacyjnych [1, 4, 11, 18, 25, 27] stanowią: 1) synteza układu, polegająca na określeniu minimalnej normalnej postaci funkcji przełączającej, 2) wybór realizacji logicznej syntetyzowanego układu. Podkreślić należy znaczny rozwój teorii progowych układów kombinacyjnych [3, 8, 21, 27, 29].

W teorii synchronicznych automatów zdeterminowanych konwencjonalnych [12, 14, 15, 17, 22, 27, 28, 30, 31, 33, 34, 35] rozwinęły się i ugruntowały algebraiczne podstawy tej teorii. Odgrywają one istotną rolę w problemach: równoważności automatów, dekompozycji, kodowania stanów, realizacji automatów.

W problematyce asynchronicznych automatów zdeterminowanych szczególnie ważne są zagadnienia: hazardu i gonitwy [27, 37].

Badanie relacji: automat zdeterminowany — język, było przedmiotem zainteresowania badaczy przez dłuższy okres czasu. Szczytowymi osiągnięciami w tej dziedzinie są centralne twierdzenia analizy i syntezy [2, 23, 27]. Uwaga specjalistów skupia się na pracach z zakresu podstaw lingwistyki matematycznej, np. [5].

Teoria automatów stochastycznych konwencjonalnych [26, 27, 32, 35] rozwija się szczególnie dynamicznie w ostatnich latach. Zainteresowania badaczy koncentrują się na problemach języków w tych automatach. W ramach teorii automatów specjalnych szczególnie ciekawe rezultaty osiągnięto w badaniach: 1) liniowych automatów zdeterminowanych [38], 2) liniowych automatów stochastycznych [7], 3) automatów zdeterminowanych zmiennych w czasie [13], 4) automatów stochastycznych z czasem ciągłym [6, 20].

Najbardziej złożoną strukturę — spośród rozpatrywanych — reprezentują automaty wolnostrukturowe dendrytowe, będące przedmiotem intensywnych badań w ciągu ostatnich kilku lat [10, 19, 24].

3. Algebraiczny opis niektórych systemów przełączających

Algebraiczny opis systemów przełączających stanowi głęboki formalizm matematyczny, umożliwiający szczegółowe wniknięcie w mechanizmy działania poszczególnych systemów przełączających. W rozważaniach zamieszczonych w niniejszym punkcie dokonuje się algebraicznego opisu niektórych klas systemów przełączających. Opis ten uwidacznia ewolucję wewnętrznej struktury systemów przełączających: od układów kombinacyjnych konwencjonalnych, stanowiących najprostsze systemy przełączające, do automatów wolnostrukturowych, mających najbardziej rozbudowaną meta-strukturę.

3.1. Układy kombinacyjne

3.1.1 Układy kombinacyjne konwencjonalne (U.K.K)
Definicja U.K.K

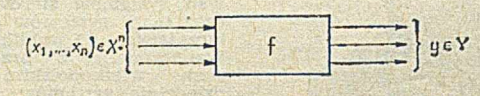
$$\text{U.K.K.} = \langle X, Y, f \rangle, \qquad 3.1$$

gdzie:

$X = \{0,1\}$ — alfabet wejściowy,
$Y = \{0,1\}$ — alfabet wyjściowy
$f: X^n \to Y$ — funkcja przełączająca.

$(x_1,..,x_n) \in X^n$, gdzie $x_i \in X$, nazywamy n-pozycyjną binarną kombinacją wejściową, lub słowem wejściowym o długości n.

Schemat blokowy U.K.K. przedstawiony jest na rysunku 3.1.



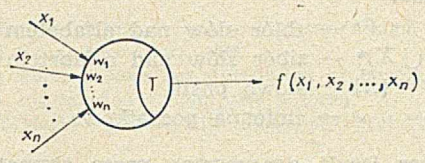
Rys. 3.1

Rys. 3.1 Schemat blokowy układu kombinacyjnego konwencjonalnego

3.1.2. Układy kombinacyjne progowe (U.K.P.)
Definicja U.K.P.

$$\text{U.K.P.} = <X, Y, W, +, ., \geqslant, <, f, T, >, \qquad 3.2$$

przy czym:



Rys. 3.2.

Rys. 3.2 Schemat blokowy układu kombinacyjnego progowego

$$\forall\,[(x_1, \ldots, x_n) \in X^n \,|\, f(x_1, \ldots, x_n) = 1] \qquad \sum_{i=1}^{n} w_i x_i \geqslant T$$

oraz

$$\forall\,[(x_1, \ldots, x_n) \in X^n \,|\, f(x_1, \ldots, x_n) = 0] \qquad \sum_{i=1}^{n} w_i x_i < T.$$

W definicji tej oznaczają:
$W = \{w_1, \ldots, w_n\}$ — zbiór wag,
T — próg układu kombinacyjnego progowego.

Pozostałe oznaczenia: jak w definicji 3.1
Schemat blokowy U.K.P. zamieszczony jest na rysunku 3.2

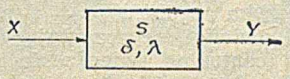
3.2. Automaty zdeterminowane konwencjonalne

3.2.1. Automaty zdeterminowane synchroniczne (A.z.s.)
Definicja A.z.s.

$$\text{A.z.s.} = <S, X, Y, \delta, \lambda, > \qquad 3.3$$

gdzie:

S — niepusty zbiór stanów,
X — niepusty alfabet wejściowy,
Y — niepusty alfabet wyjściowy,
$\delta : (S \times X)_t \to S_{t+1}$ — funkcja przejść,
$\lambda : (S \times X)_t \to Y_t$ — funkcja wyjść (dla automatów klasy Mealy'ego)



Rys. 3.3.

Rys. 3.3 Schemat blokowy automatu zdeterminowanego synchronicznego

Schemat blokowy A.z.s. przedstawiony jest na rysunku 3.3.

3.2.2. Automaty zdeterminowane asynchroniczne (A.z.as.)
Definicja A.z.as.

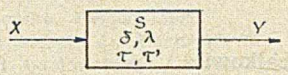
$$\text{A.z.as} = <S, , X, Y, \delta, \lambda, \tau, \tau' >, \qquad 3.4$$

przy czym:

$$s(t + \tau) = \delta[s(t), x(t)],$$
$$y(t + \tau') = \lambda[s(t), x(t)],$$

gdzie τ i τ' oznaczają czas wykonywania operacji przez automat. (Czas τ' bywa w rozważaniach teoretycznych na ogół pomijany).

Pozostałe oznaczenia: jak w definicji 3.3.
Schemat blokowy A.z.as. zilustrowany jest na rysunku 3.4.



Rys. 3.4.

Rys. 3.4 Schemat blokowy automatu zdeterminowanego asynchronicznego

3.3. Automaty stochastyczne konwencjonalne (A.st.)
Definicja A.st.

$$\text{A.st} = <S, X, F, \{M(x)\}, \pi_0, > \qquad 3.5$$

gdzie:

F — zbiór dopuszczalnych stanów końcowych ($F \subseteq S$),
$\{M(x)\}$ — zbiór stochastycznych macierzy przejść,
π_0 — stochastyczny stan początkowy.

Pozostałe oznaczenia: jak w definicji 3.3.
Zbiór dopuszczalnych stanów końcowych F scharakteryzowany jest przez zbiór (macierz $n \times 1$) n binarnych wielkości.
Odpowiednio:

$$\mu = \begin{cases} 1, & \text{gdy } s_i \in F \\ 0, & \text{gdy } s_i \in F \end{cases} \quad i = 1, \ldots, n \qquad 3.6$$

Stochastyczna macierz przejść określona jest następująco:

$$M(x) = [m_{ij}]^{n \times n}, \qquad 3.7$$

gdzie:

$m_{ij} = m_{ij}(s_j | s_i, x)$ $(0 \leqslant m_{ij} \leqslant 1)$ określa prawdopodobieństwo przejścia stanu s_i do stanu s_j pod działaniem symbolu x, zaś $n = |S|$. Oczywiście zachodzi związek:

$$\sum_{i=1}^{n} m_{ij} = m_j = 1. \qquad 3.8$$

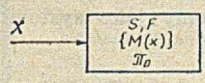
π_0 określa rozkład prawdopodobieństwa na zbiorze stanów S w chwili początkowej:

$$\pi_0 = \begin{pmatrix} s_1, \ldots, s_n \\ p_1 \ldots, p_n \end{pmatrix}, \qquad 3.9$$

przy czym zachodzi zależność:

$$\sum_{i=1}^{n} p_i = 1. \qquad 3.10$$

Schemat blokowy automatu stochastycznego zamieszczony jest na rysunku 3.5.



Rys. 3.5.

Rys. 3.5 Schemat blokowy automatu stochastycznego.

3.4. Automaty wolnostrukturowe

3.4.1. Automaty typu „sinking" (opadające) (A.w.s.)
Definicja A.w.s.

$$\text{A.w.s.} = < S, X, F, \delta >, \qquad 3.11$$

gdzie:

F — zbiór dopuszczalnych stanów końcowych,
$\delta : S[\mu] \times X \to P(S)$,

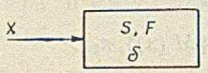
przy czym:

μ — liczba całkowita,

$S[\mu] = \bigcup_{\nu=0}^{\mu} S^{\nu}$,

$P(S)$ — zbiór wszelkich podzbiorów zbioru S.

Pozostałe oznaczenia: jak w definicji 3.3.
Schemat blokowy A.w.s. zobrazowany jest na rysunku 3.6.



Rys. 3.6 Schemat blokowy automatu wolnostrukturowego typu „sinking"

3.4.2. Automaty typu „climbing" (wznoszące) (A.w.c.)
Definicja A.w.c.

$$\text{A.w.c.} = < S, X, F, \Delta >, \qquad 3.12$$

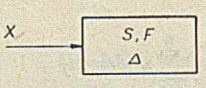
gdzie:

F — zbiór dopuszczalnych stantów końcowych,
$\Delta : [S \cup \{\Lambda\}] \times X \to P(S^*)$,

przy czym:

Λ — słowo puste, element monoidu S^*,
$P(S^*)$ — zbiór wszystkich podzbiorów monoidu S*,

Pozostałe oznaczenia: jak w definicji 3.3.
Schemat blokowy A.w.c. pokazany jest na rysunku 3.7.



Rys. 3.7 Schemat blokowy automatu wolnostrukturowego typu „climbing"

4. Języki w systemach przełączających

Teoria systemów przełączających jest teorią komplementarną, obejmującą zarówno aspekty systemowe jak i językowe. W ramach analizy podejmujemy badania, dotyczące rozpoznawania przez a priori istniejące systemy przełączające (automaty) języków wejściowych (problemy akceptowalności języków), natomiast w ramach syntezy dokonujemy konstrukcji systemu przełączającego, reprezentującego a priori określony język wejściowy (problemy reprezentowalności).
Do określenia języka konieczna jest znajomość alfabetów: nieterminalnego (V) i terminalnego (X) oraz gramatyki (G).
Gramatykę G definiujemy jako system:

$$G = < V, X, P, s >, \qquad 4.1$$

gdzie:

V — zbiór symboli (zmiennych) nieterminalnych (alfabet nieterminalny),
X — alfabet terminalny, $X \cap V = \Phi$,
P — skończony zbiór uporządkowanych par (produkcji), (φ, ψ), $(\varphi \to \psi)$,

przy czym:

$$\varphi \in V^* - \{\Lambda\}, \quad \Psi \in (V \cup X)^*,$$

gdzie:

V^* — zbiór słów nad alfabetem V,
$(V \cup X)^*$ — zibór słów nad alfabetem $(V \cup X)$,
$\{\Lambda\}$ — słowo puste,
s — zmienna początkowa.

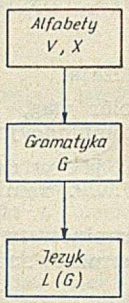
Przez język, generowany przez gramatykę G, rozumiemy zbiór słów $L(G)$:

$$L(G) = \{x^* \mid x^* \in X^*, \ s \to \underset{\otimes}{} x^*\}, \qquad 4.2$$

przy czym:

$s \to \underset{\otimes}{} x^*$ oznacza, że istnieje ciąg: $\varphi_0, \varphi_1, \ldots, \varphi_i, \ldots, \varphi_n$, taki, że produkcja $s \to \underset{\otimes}{} x^*$ ma miejsce ($\psi_1 = \varphi_0 = s$, $\psi_2 = \varphi_n = x^*$).

Procedura generacji języka $L(G)$ pokazana jest blokowo na rysunku 4.1.



Rys. 4.1 Schemat procedury generacji języka L(G)

N. Chomsky zaproponował pewną klasyfikację gramatyk [5], która ma duże znaczenie w badaniach akceptowalności i reprezentowalności języków przez automaty.
Załóżmy, że $L \subseteq X$. Można pokazać [2, 23, 27], że zachodzą dwa centralne twierdzenia, zwane twierdzeniami analizy i syntezy:
1. Jeżeli automat zdeterminowany A.z. akceptuje język L, to język ten jest regularny (centralne twierdzenie analizy),
2. Dla każdego języka regularnego — L reg. — można skonstruować odpowiedni, reprezentujący go automat zdeterminowany (centralne twierdzenie syntezy).
Szczególnie interesujące są badania w zakresie rozpoznawania języków przez automaty stochastyczne [20] i automaty wolnostrukturowe [19].

5. Możliwości technicznych zastosowań modeli systemów przełączających

Formalizmy matematyczne, opisujące modele systemów przełączających, mogą być wykorzystane, dzięki ich ogólności, w badaniach innych systemów oraz analizie konkretnych modeli technicznych. Zakres możliwych zastosowań ulega ciągłemu poszerzaniu. Przykładowo, można tutaj wskazać modele dyskretnych kanałów informacyjnych oraz modele niezawodnościowe. Pierwsze z nich można potraktować jako liniowe automaty stochastyczne [16], drugie, jako automaty stochastyczne z czasem ciągłym [9, 36].
Szczególnie duże możliwości wykorzystania algebraicznego aparatu pojęciowego teorii systemów przełączających oraz jej rezultatów istnieją w dziedzinie dyskretnych systemów sterowania.

Literatura

[1] Bromirski J.: Teoria automatów.
Wydawnictwa Naukowo-Techniczne, Warszawa 1969.

[2] Brzozowski J. A.: Derivates of regular expressions.
Journal of A.C.M., vol. 11 No 4 (October 1964).

[3] Butakow E. A., Zakriewskij A. D.: Niekotoryje woprosy realizacji buliewych funkcji porogowymi eliementami.
Izwiestia A.N. ZSRR, „Tiechniczeskaja Kibernetika" No. 1, 1964.

[4] Caldwell S. H.: Switching circuits and logical design.
John Wiley and Sons, New York 1958.

[5] Chomsky N.: Three models for description of language.
I.R.E. Trans. on Information Theory, No 2 (1956).

[6] Chung K. I.: Markov chains with stationary probabilities.
Springer, Berlin 1960.

[7] Cohn M.: Properties of linear machines.
Journal of A.C.M. 11 (1964).

[8] Dertouzos M. L.: Threshold logic: a synthesis approach.
M.I.T. Press, Cambridge, Mass. 1965.

[9] Dolazza E.: System state analysis and flow graphs diagrams in reliability.
I.E.E.E. Trans Reliability R-15, No 3, 1966, str. 85—94.

[10] Doner J. E.: Tree acceptors and some of their applications.
Syst. Devel. Corp. Science Report No 8, 1967.

[11] Gawriłow M. A.: Tieoria rielejnokontaktnych schiem.
Izd. A.N. ZSRR, 1950.

[12] Gill A.: Introduction to the theory of finite state automata machines.
Mc Graw-Hill Book Comp., New York 1962.

[13] Gill A.: Time-varying sequential machines.
Journal Franklin Inst. 276 (Dec. 1963), str. 519—539.

[14] Harrison M. A.: Introduction to switching and automata theory.
Mc Graw-Hill Book Comp., New York 1965 (również tłum. polskie): Wstęp do teorii sieci przełączających i teorii automatów, PWN, Warszawa 1973).

[15] Hartmanis J., Stearns R. E.: Algebraic structure theory of sequential machines.
Prentice-Hall Inc. Englewood Cliffs, New York 1964.

[16] Howard R.: Dynamic programming and Markov processes.
M.I.T. Press and J. Wiley, New York 1970.

[17]Huffman D. A.: The synthesis of sequential switching circuits.
Journal Franklin Inst., t. 257, nr 3 (1954).

[18] Humphrey W. S. Jr.: Switching circuits with computer applications.
Mc Graw-Hill Book Comp., New York 1958.

[19] Karpiński M.: Free structure tree automata.
Rozprawa doktorska, Poznań 1973.

[20] Knast R.: Modele matematyczne i elektryczne automatów stochastycznych.
Rozprawa doktorska, Poznań 1969.

[21] Lisiecka-Frąszczak J., Puchałka T.: O możliwościach syntezy funkcji boolowskich na elemencie progowym o zmiennym progu.
Prace VI Krajowej Konferencji Automatyki, t. I, Poznań 1974, str. 213—222.

[22] Mc Cluskey E. I.: Introduction to the theory of switching circuits.
Mc Graw-Hill Book Comp., New York 1964.

[23] Mc Naughton R., Yamada H.: Regular expressions and state graphs for automata.
I.R.E. Trans. EC — 9, 1960.

[24]Magidor M., Moran G.: Probabilistic tree automata and contex free languages.
Israel Journal Math. No 8 (1970).

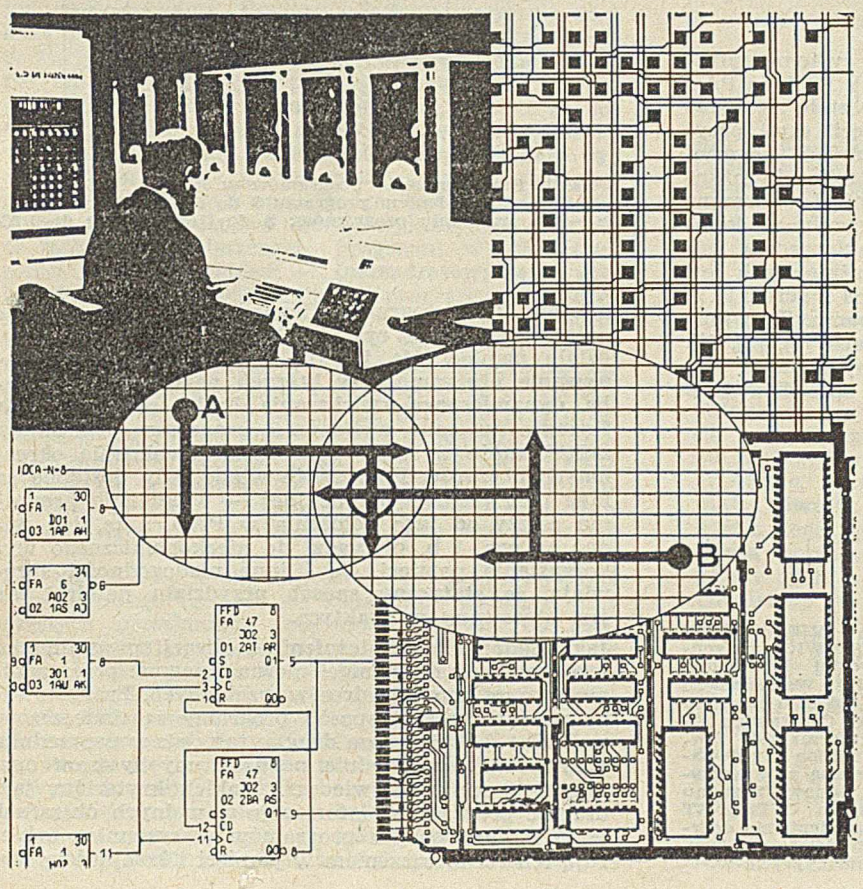
[25] Moisil G.: Algiebraiczeskaja tieoria diskretnych awtomaticzeskich ustrojstw. (Tłum. z jęz. rumuńskiego).
Izd. Innostr. Lit., 1962.

[26] Paz A.: Introduction to probabilistic automata.
Academic Press, New York 1971.

[27] Puchałka T.: Teoria systemów przełączających.
Wykłady na studium doktoranckim w zakresie automatyki i informatyki, prowadzone w roku akad. 1973/74.
Skrypt uczelniany, AGH, Kraków, w druku.

[28] Puchałka T.: Systemy przełączające. Zagadnienia reprezentacji i rozpoznawania języków. Modele pewnych zastosowań technicznych.
Materiały II Ogólnopolskiego Sympozjum nt.: „System — Modelowanie — Sterowanie". Zakopane 1974, str. 97.

[29] Puchałka T., Lisiecka J., Siwak P.: Niektóre problemy syntezy sieci elementów progowych.
Prace IV Krajowej Konferencji Automatyki, t. II, AGH, Kraków, 1967, str. 93—106.



[30] Puchałka T., Mikołajczak B.: O optymalnym kodowaniu stanowym pewnych klas synchronicznych maszyn sekwencyjnych.
Prace V Krajowej Konferencji Automatyki, t. 2, Politechnika Gdańska, Gdańsk 1971, str. 79—85.

[31] Puchałka T., Woźniak A., Grzymała-Busse J.: Generalised synthesis of finite automata by algebraic methods.
Materiały III Kongresu IFAC 1966, W. Clowes and Sons Limited, London and Beecles, str. 35F1—35F7.

[32] Rabin M. O.: Probabilistic automata.
Information and Control 6 (1963).

[33] Rabin M. O., Scott D.: Finite automata and their decision problems.
IBM, 1959.

[34] Salomaa A.: Theory of automata.
Pergamon Press, 1969.

[35] Starke P. H.: Abstrakte automaten.
VEB Deutscher Verlag der Wissenschaften, Berlin 1968.

[36]Tin Htun L.: Reliability prediction techniques of complex systems.
I.E.E.E. Trans. Reliability R-15, 1966, str. 58—69.

[37] Torng H. C.: Introduction to the logical design of switching systems.
Addison-Wesely Publishing Comp. Inc., 1964.

[38] Yau S. S., Wang C. K.: Linearity of sequential machines.
I.E.E.E Trans. Electr. Comp. EC — 15 (1966).

MIECZYSŁAW MURASZKIEWICZ
Centralny Ośrodek Informatyki
Politechnika Warszawska

681.327.2:681.322.001.1

Przegląd wybranych metod wykorzystywania i organizacji pamięci systemu cyfrowego

W artykule przedstawiono trzy metody wykorzystania i organizacji pamięci systemu cyfrowego tworzące ciąg ewolucyjny: przeadresowywanie, stronicowanie i segmentację. Wskazano również na główne wady i zalety każdej z nich. Ponadto podano koncepcje maszyny wirtualnej będącej zwieńczeniem tego ciągu ewolucyjnego.

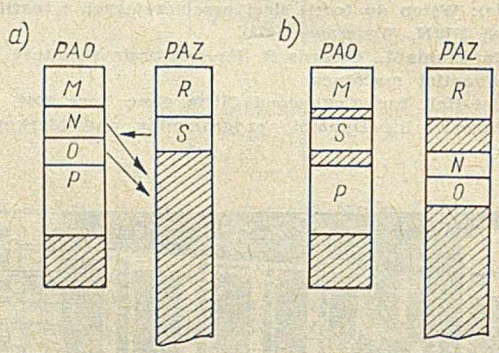
Poszukiwanie sposobów lepszego wykorzystania jednostki centralnej systemu cyfrowego doprowadziło do opracowania takich koncepcji, jak: przetwarzanie wsadowe, wieloprogramowanie i praca z podziałem czasu. Charakteryzują się one między innymi tym, że w pamięci systemu może jednocześnie znajdować się więcej niż jeden program. Zazwyczaj jednak programy przeznaczone do wykonania w procedurze są na tyle długie, bądź ich liczba jest tak duża, że nie można ich wszystkich przechowywać w pamięci operacyjnej systemu (dalej PAO). W takich przypadkach część tych programów lub ich fragmenty są umieszczane w pamięci zewnętrznej procesora (dalej PAZ). Pociąga to jednak za sobą konieczność przepisywania tych programów do właściwych obszarów PAO zawsze wtedy, gdy mają być one realizowane w procesorze.

Obecność wielu programów w pamięci systemu cyfrowego oraz konieczność ich przesyłania z PAZ do PAO lub odwrotnie sprawiły, że **nieodzowne stało się opracowanie metod tzw. przydziału** pamięci, czyli przyporządkowania różnym obszarom pamięci poszczególnych programów i danych [1]. Realizacją czynności związanych z przydziałem pamięci obarczono odpowiednie programy systemu operacyjnego komputera. Programy te powinny charakteryzować się określonymi cechami, a jedną z najważniejszych jest ochrona programów nieaktywnych przed ingerencją ze strony aktualnie wykonywanego programu. Ta i inne cechy zostały omówione m.in. w pracach [2], [3], [4]. Poniżej, poczynając od historycznie najstarszej i jednocześnie koncepcyjnie najprostszej, przedstawiono różne metody przydziału pamięci.

Mgr inż. MIECZYSŁAW MURASZKIEWICZ ukończył Wydział Elektroniki Politechniki Warszawskiej w 1972 r. Od tego też roku pracuje w Centralnym Ośrodku Informatyki PW. Zajmuje się on problematyką związaną z wykorzystaniem i budową Systemu Abonenckiego CYFRONET oraz zagadnieniami dotyczącymi translatorów języków symulacyjnych.

Przeadresowywanie

Najprostszy sposób przydziału pamięci, tzw. **statyczny** polega na tym, że programy (przetłumaczone i zapisane przy użyciu adresów rzeczywistych) są umieszczane w PAO sekwencyjnie (jeden za drugim). Mogą być one jednak wykonywane w innej kolejności — decyduje o tym system operacyjny. Również programy znajdujące się w PAZ są zapisane przy użyciu adresów rzeczywistych pamięci operacyjnej. Oznacza to, że realizacja programu znajdującego się w PAZ musi być poprzedzona zwolnieniem żądanego obszaru pamięci operacyjnej oraz przesłaniem w to

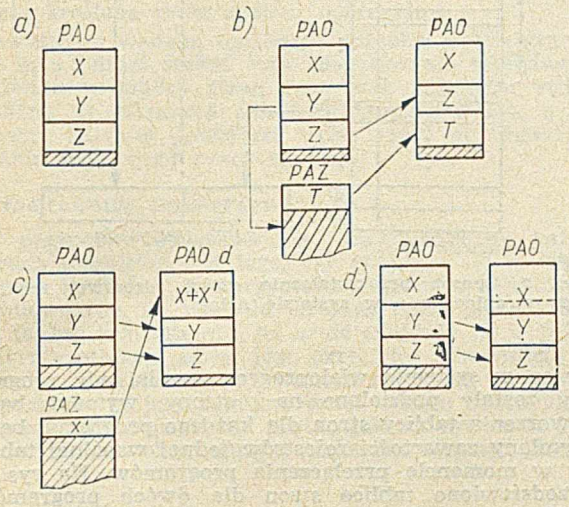


Rys. 1. Rozmieszczenie programów w PAO i PAZ
a — przed przesłaniem programu S do PAO;
b — po przesłaniu programów: S do PAO, N i Q do PAZ

miejsce tego programu. Czynności te schematycznie przedstawiono na rys. 1. Przy tym sposobie przydziału pamięci **nie jest w zasadzie możliwe wykonywanie programów, w których pojawia się potrzeba dynamicznego zwiększania liczby pól roboczych.** Ponadto w celu wykonania programu przesłanego z PAZ może wystąpić konieczność zwolnienia określonych obszarów PAO także wtedy, gdy dostępne są inne pola pamięci operacyjnej — wykonanie programu zależy od jego położenia w PAO, a to, jak widać na rys. 1 b, prowadzi do nieekonomicznego wykorzystania pamięci. Te i inne niedogodności sprawiają, że **statyczny sposób przydziału pamięci nie jest stosowany w praktyce.**

Jak wiadomo, uniezależnienie realizacji programu od jego położenia w pamięci można osiągnąć przez zapisanie go za pomocą adresów względnych. Przygotowane właśnie w taki sposób programy są umieszczone w PAO i PAZ jeden za drugim tak, jak w poprzednim przypadku. Odpowiednie podprogramy systemu operacyjnego na podstawie m.in. takich kryteriów, jak: długość programów, informacje w wolnych obszarach PAO i czasy pobytu programów w systemie — kierują ich rozmieszczeniem w pamięci i kolejnością wy-

konywania. Jednakże i ten sposób przydziału pamięci, tzw. **dynamiczny** nie jest wolny od wad, wśród których najuciążliwsze są zilustrowane schematycznie na rys. 2. W sposobie tym wyeliminowana została natomiast istotna wada statycznego przydziału pamięci, a mianowicie — jej nieracjonalne wykorzystanie.



Rys. 2. Przydział pamięci dla trzech programów X, Y, Z przy pracy wieloprogramowej
a — rozmieszczenie programów X, Y, Z w pamięci;
b — przesuwanie wynikające z usunięcia programu Y z PAO i rozpoczęcia programu T;
c — przesuwanie wynikające z uzupełnienia programu X;
d — przesuwanie wynikające z rozszerzenia się pól roboczych programu X

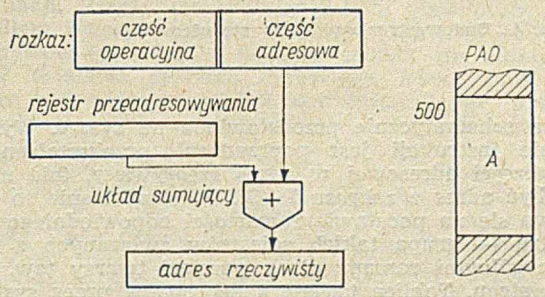
Ustalone przez system operacyjny położenie programu w pamięci w zasadzie nie zmienia się w czasie jego wykonywania. Ponieważ jednak wszystkie rozkazy programu — w przypadku dynamicznego przydziału pamięci — mają części adresowe takie, jakby był on umieszczony od pewnego umownego miejsca pamięci (zazwyczaj o numerze zero), przeto zachodzi konieczność zmiany zawartości części adresowych większości instrukcji. **Modyfikacja adresów może być realizowana bądź programowo** — przez specjalny podprogram systemu operacjnego, **bądź układowo** [5]. Każdy z tych sposobów można wykorzystać albo już w momencie umieszczania programu w PAO (tzn. przed jego wykonaniem — **(przeadresowywanie globalne)**, albo w trakcie realizacji programu **(przeadresowywanie dynamiczne)**. Zaletą tego drugiego sposobu jest to, że nie są zmieniane adresy tych rozkazów, które w trakcie wykonywania programu nie będą z jakichś powodów realizowane, co oczywiście przyspiesza wykonanie programu. Zarówno przy przeadresowywaniu globalnym, jak i dynamicznym, modyfikowane są adresy tylko tych instrukcji, które zostały na podstawie ich treści opatrzone specjalną etykietą w czasie trwania translacji programu. Unika się dzięki temu wydłużania procesu jego realizacji.

Ze względu na małą sprawność przeadresowywania programowego co staje się szczególnie widoczne przy częstych przesłaniach programów z PAZ do PAO oraz licznych przełączeniach między programami, nie znalazło ono powszechnego zastosowania w praktyce. Znacznie częściej korzysta się z **przeadresowywania układowego**, dla którego zarówno w przypadku przeadresowywania globalnego jak i dynamicznego stosuje się tzw. **rejestr przeadresowywania** nazywany inaczej rejestrem bazowym. Zawartość tego rejestru jest kontrolowana przez specjalny podprogram systemu operacyjnego i ulega zmianie w zależności od aktualnie wykonywanego programu. Mechanizm przeadresowywania umożliwiający uzyskanie adresu rzeczywistego przedstawiono schematycznie na rys. 3 — modyfikowany adres jest dodawany w układzie sumującym do zawartości rejestru przeadresowywania dając w ten sposób adres rzeczywisty.

Poprawę sprawności tego sposobu przeadresowywania można osiągnąć przez zastosowanie kilku rejestrów, z których każdy jest związany z innym programem znajdującym się w PAO. W takim przypadku przy przełączaniu między programami nie trzeba już zmieniać zawartości rejestrów, co wydatnie poprawia sprawność działania systemu, a ponadto — po pewnej rozbudowie systemu operacyjnego — pozwala na przechowywanie programów w postaci rozłącznych bloków.

Możliwość przechowywania programu w postaci rozłącznych bloków jest o tyle cenna, iż pozwala lepiej i sprawniej wykorzystać pamięć operacyjną, zwłaszcza wtedy, gdy bloki te są stosunkowo niewielkich rozmiarów. Wówczas można nimi wypełnić „luki" pomiędzy programami znajdującymi się w PAO bez konieczności przesuwania czy odsyłania określonych programów do PAZ. Opłacone to jest jednak tym, że z każdym blokiem programu muszą być w takim przypadku stowarzyszone trzy rejestry: jeden rejestr przeadresowywania oraz dwa **rejestry**, tzw. **graniczne**, przechowujące informacje o długości bloku.

W pierwszym z nich znajduje się adres pierwszej, a w drugim ostatniej instrukcji należącej do bloku. Wykonanie większości rozkazów jest poprzedzone porównaniem ich części adresowych z zawartością kolejnych par rejestrów granicznych, co trwa tak długo, aż zostanie znaleziony właściwy blok. Dopiero



Rys. 3. Przekształcenie adresu przy wykorzystaniu rejestru przeadresowywania (w dalszych rysunkach układ sumujący będzie oznaczany symbolicznie kropką opatrzoną znakiem plus)

wtedy realizowana jest modyfikacja adresu rozkazu, po czym następuje jego wykonanie. W przypadku, gdy blok taki nie zostanie odnaleziony, powinna być sygnalizowana próba ingerencji ze strony programu aktywnego w treść innych programów znajdujących się w PAO, czemu nadto musi towarzyszyć przerwanie wykonywania tego programu. Tak więc rejestry graniczne, oprócz swojej podstawowej funkcji, **spełniają również rolę układów ochrony pamięci.** Słabością tego sposobu przeadresowywania jest duża liczba porównań adresu z zawartościami rejestrów granicznych w tych przypadkach, gdy program został podzielony na wiele bloków.

Wady przedstawionych wyżej sposobów przeadresowywania, do których należy dołączyć trudności, jakie pojawiają się przy próbach takiego zorganizowania procesu przetwarzania informacji, by wiele programów mogło korzystać ze wspólnych procedur lub zbiorów danych sprawiają, że **sposoby te nie są w zasadzie wykorzystywane we współczesnych systemach cyfrowych.**

Metodą pozwalającą usunąć wiele spośród wymienionych wad jest stronicowanie.

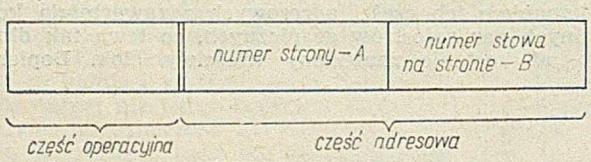
Stronicowanie

U podstaw koncepcji stronicowania (ang. *paging*) leży pomysł, by **program wraz z danymi został rozbity na grupę rozłącznych części, a adresy jego instrukcji bezpośrednio wskazywały właściwe rejestry przeadresowywania.**

Zrealizowanie stronicowania wymaga spełnienia trzech warunków. Po pierwsze należy dokonać podziału pamięci systemu cyfrowego na równe obszary (co do liczby jednostek maszynowych np. słów lub bajtów). Obszary te są nazywane **ramami stron** (ang. *page frame*). Następnie, przetłumaczony program użytkownika trzeba bez jego udziału podzielić na tzw. strony (ang. *page*), czyli takie części z których każda nie zawiera więcej jednostek niż rama strony*). I wreszcie — należy zapewnić sposób jednoznacznego przyporządkowania strony do określonej ramy strony. Długości ram stron są stałe dla danego komputera i w zależności od jego typu wahają się od 64 do 1024 słów [4].

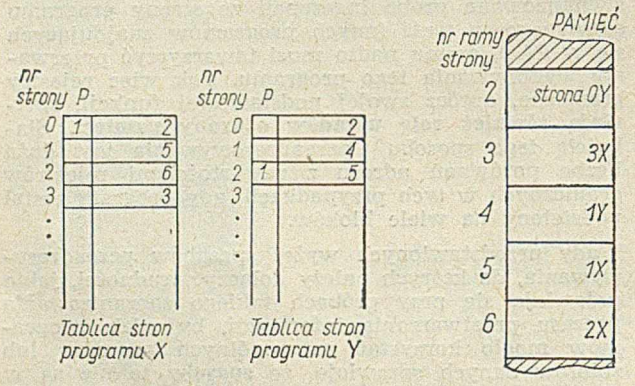
Jednakże w niektórych systemach cyfrowych — tak jak np. w systemie MULTICS [6], dopuszcza się istnienie ram stron o dwóch różnych rozmiarach.

Adres rozkazu w programie podlegającym stronicowaniu składa się z dwóch części: numeru strony i numeru słowa na stronie. Budowa takiego rozkazu zo-



Rys. 4. Budowa rozkazu przy stronicowaniu.

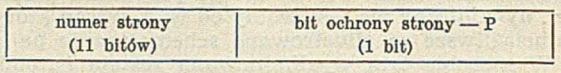
stała schematycznie przedstawiona na rys. 4. Wykonanie instrukcji jest poprzedzone przekształceniem jej części adresowej w adres rzeczywisty, (nie musi to być adres z zakresu PAO). Przekształcenie to odbywa się na podstawie zawartości odpowiedniego rejestru — grupa takich rejestrów związanych z poszczególnymi ramami stron pamięci tworzy tzw. **tablicę stron.** Tablica ta jest nadzorowana przez system operacyjny. W przypadku, gdy wyznaczony adres rzeczywisty wskazuje komórkę pamięci znajdującą się w PAZ, wówczas do PAO jest przepisywana odpowiednia strona. Zasady, według których odbywa się transmisja stron opisują specjalne algorytmy. Dokładne informacje na ten temat można znaleźć w pracach [7] i [4].

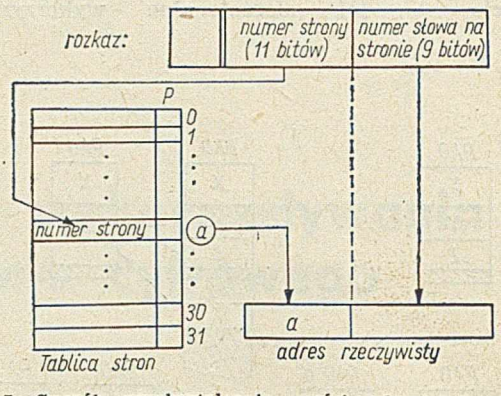


Rys. 5. Zawartość tablicy stron dwóch programów X i Y oraz obraz przydzielonej im pamięci

*) Niekiedy stronami są nazywane również ramy stron.

Rys. 6





Rys. 7. Sposób przekształcania części adresowej rozkazu przy stronicowaniu w systemie Atlas

Praca w systemie wieloprogramowania, gdy programy zostały podzielone na strony, wymaga bądź utworzenia tablicy stron dla każdego programu, bądź wymiany zawartości rejestrów jednej wspólnej tablicy w momencie przełączenia programów. Na rys. 5 przedstawiono tablice stron dla dwóch programów X i Y oraz obraz pamięci przydzielonej im przez system operacyjny.

Odwołanie od odpowiedniego słowa w tablicy stron jest realizowane przez podanie numeru strony A przekształcanego adresu. Zawartość tego słowa wskazuje numer ramy strony w pamięci (w rzeczywistości jest to adres pierwszej komórki tego obszaru). Jedynka w polu P słowa w tablicy stron oznacza, że dana strona jest niedostępna w pamięci.

Widoczną już obecnie **zaletą stronicowania jest stosunkowo prosto zrealizowana możliwość nieciągłego rozmieszczania programów w pamięci** co znacznie podnosi efektywność jej wykorzystania — niepotrzebne jest więc przesuwanie pól informacyjnych w przypadkach występujących w omówionym już przeadresowywaniu.

Stronicowanie jednopoziomowe

Opisany poniżej klasyczny już sposób stronicowania, zwany jednopoziomowym, został opracowany dla komputerów systemu ATLAS [8], [9] i jako jeden z pierwszych był realizowany w praktyce. Z każdą ramą strony w PAO tego systemu jest stowarzyszony jeden rejestr, w którym przechowuje się (w sposób podany na rys. 6) następujące informacje:

Jedenaście bardziej znaczących bitów tego rejestru określa numer strony, a zatem w systemie ATLAS można wykorzystać 2048 stron. Rama strony zaś liczy 512 słów maszynowych. Rejestry związane z ramami stron w PAO są zgrupowane w tablicę stron, która składa się z 32 słów (rejestrów).

Sposób przekształcania części adresowej rozkazu w adres rzeczywisty przedstawiono schematycznie na rys. 7.

Część adresowa rozkazu jest podzielona na dwie grupy: 11 bardziej znaczących bitów wskazuje numer strony, a tym samym — pośrednio — odpowiednie słowo w tablicy stron. Pozostałe 9 bitów określa adres komórki wewnątrz wyznaczonej strony. Możliwe są dwa przypadki w procesie przekształcania adresu, gdy numer żądanej strony:

- zostanie odszukany w którymś rejestrze w tablicy stron. Wówczas adres rzeczywisty jest tworzony poprzez złożenie numeru tego rejestru z adresem komórki na danej stronie. Nie występuje tu dodawanie adresów, tak jak w przypadku użycia rejestru przeadresowywania, a raczej sumowanie logiczne.

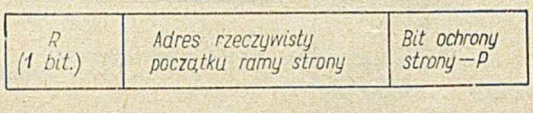
● nie jest zapisany w żadnym rejestrze, co oznacza, że strona znajduje się w PAZ. W takim przypadku jest ona przesyłana do PAO, czemu towarzyszy wymiana zawartości określonego rejestru z tablicy stron. Dalsze operacje są takie same jak w pierwszym przypadku.

Bit ochrony strony jest wykorzystywany po to, by z aktualnie wykonywanego programu umożliwić (P = = 1) odwołanie do pewnych stron, np. wtedy, gdy są to strony innego programu. Wartość tego bitu jest określona przez system operacyjny.

Poważnymi wadami opisanego sposobu stronicowania są m.in.: **długi średni czas odszukiwania właściwego rejestru w tablicy stron** oraz **brak rejestrów związanych ze stronami umieszczonymi w PAZ**, a to jeszcze bardziej powiększa straty czasu na organizowanie poprawnej pracy systemu.

Stronicowanie standardowe

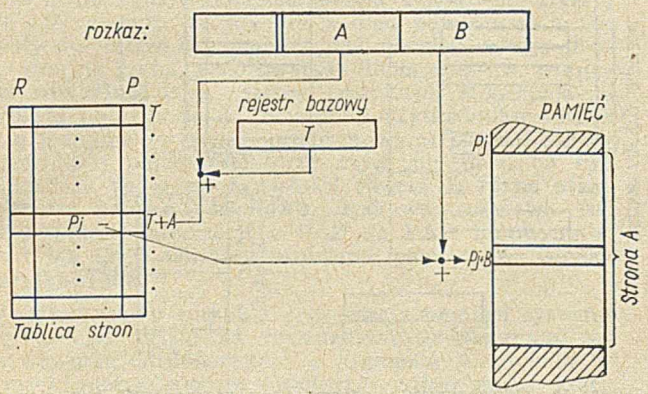
W przypadku stronicowania standardowego każda strona niezależnie od tego, czy znajduje się w PAO, czy w PAZ jest stowarzyszona z jednym słowem w tablicy stron. Ze względu na dużą liczbę słów, tablica ta, lub przynajmniej jej część znajduje się w PAO. Jej początkowy adres jest określony zawartością T specjalnego rejestru, nazywanego rejestrem bazowym. Zawartość ta jest ustalana przez system operacyjny. Słowo tablicy stron ma postać jak na rys. 8, R jest wskaźnikiem określającym położenie strony: R = 1,



Rys. 8. Postać słowa tablicy stron

gdy znajduje się ona w PAO; R = 0, gdy strona jest umieszczona w PAZ. Sposób odwzorowania części adresowej rozkazu w adres rzeczywisty przedstawiono schematycznie na rys. 9. Adres rzeczywisty jest tworzony z adresu początku ramy strony P_j pochodzącego z określonego wcześniej słowa tablicy stron oraz numeru słowa B na danej stronie, pochodzącego z odwzorowywanego adresu. Adres słowa tablicy stron jest wyznaczony jako suma numeru strony A i zawartości rejestru bazowego T. W przypadku, gdy żądana strona znajduje się w PAZ, jest ona przepisywana do PAO. Bit P ochrony strony służy do celów, o których wspomniano przy opisie stronicowania jednopoziomowego.

Oprócz tego, że **przy stronicowaniu standardowym średni czas odwzorowania adresu wydłuża** się (operacja sumowania, porównywanie zawartości komórek, ewentualne przesłania stron z PAZ do PAO i odwrotnie) ma ono jeszcze tę wadę, że **tablica stron zajmuje znaczny obszar pamięci operacyjnej**. Z tego powodu część tej tablicy musi być przechowywana



Rys. 9. Sposób odwzorowania adresu przy stronicowaniu standardowym

w PAZ, a to dodatkowo komplikuje i wydłuża proces przekształcania adresu. Pewnym rozwiązaniem tego problemu jest **organizacja pamięci ze stronicowaniem dwupoziomowym**, w którym stronicowaniu podlega również tablica stron [5]. W takim przypadku adres jest dzielony na trzy części: A, B, C, gdzie para A, B wskazuje numer strony, zaś C adres na stronie. Sposób odwzorowania takiego adresu w adres rzeczywisty jest podobny do tego, który stosuje się w systemie organizacji pamięci zwanym segmentacją.

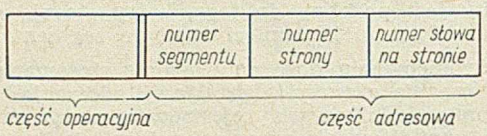
Przy stronicowaniu dość powszechnie stosowaną metodą skracania czasu przekształcania adresu jest **użycie grupy dodatkowych rejestrów tworzących niewielką pamięć asocjacyjną** o bardzo krótkim czasie dostępu [10], [11]. W komórkach tej pamięci przechowuje się — wraz z adresami rzeczywistymi początków ram stron — numery tych stron, do których odwoływano się stosunkowo niedawno, bądź były one wykorzystywane bardzo często. Przed przekształceniem adresu według metod podanych powyżej, sprawdza się czy numer żądanej strony znajduje się w jednej z komórek pamięci asocjacyjnej — jeśli tak, to proces odwzorowania znacznie się skraca, bowiem nie trzeba wykonywać operacji pośrednich związanych z przekształceniem adresu. W przeciwnym razie odwzorowanie adresu jest realizowane w sposób przedstawiony poprzednio. Pamięć asocjacyjna znajduje się pod stałym nadzorem systemu operacyjnego, który na podstawie stosunkowo prostych algorytmów ustala zawartości poszczególnych komórek. Szersze informacje na ten temat można znaleźć w pracach [8] i [12].

Wprowadzenie stronicowania miało na celu głównie osiągnięcie takich rezultatów, jak: **zmniejszenie stopnia niewykorzystania pamięci systemu cyfrowego, umożliwienie dynamicznego rozmieszczenia programów w pamięci oraz ochrona przechowywanych w niej informacji.** Praktyczne spełnienie tych postulatów jest źródłem różnego rodzaju kłopotów [13], [14], których pomyślne rozwiązanie wpływa jednak na efektywność działania systemu cyfrowego. Efektywność ta może być mierzona takimi wskaźnikami, jak: koszty rozbudowy systemu cyfrowego (szczególnie nakłady na rozszerzenie jego systemu operacyjnego), strata czasu pracy systemu wynikająca z dużej liczby transmisji stron oraz ewentualne straty pewnych obszarów pamięci. Dokładniejsze dane na temat metod mierzenia efektywności systemów cyfrowych, w których realizuje się stronicowanie oraz uzyskanych za pomocą tych metod wyników można znaleźć w pracy [4].

Segmentacja

Omówione dotychczas techniki przeadresowywania i stronicowania stanowią dwa ogniwa łańcucha wyznaczającego ewolucję sposobów przydziału pamięci. Trzecim jego ogniwem jest tzw. segmentacja (ang. *segmenting*). Polega ona na tym, że i program i dane zostają (według ich logicznej struktury) podzielone na części nazywane segmentami. **O ile przy stronicowaniu podział programu na strony odbywał się bez udziału użytkownika, o tyle w przypadku segmentacji wymaga się od niego pewnego współdziałania,** choć znane są sposoby, np. system BURROUGHS B 5000, w których podziału programu na segmenty dokonuje wyłącznie system operacyjny. Segment, (typowy rozmiar segmentu wynosi 500 słów maszynowych) nie ma określonej reprezentacji fizycznej. Odwołanie do segmentu odbywa się przez jego symboliczną nazwę, która po przetłumaczeniu programu jest zawsze liczbą. Nieco inne ujęcie segmentacji w porównaniu z przedstawionym w niniejszym artykule podano w pracy [15]. Jednak również tam można znaleźć ważne stwierdzenie, że choć stronicowanie i segmentacja są bardzo podobne pod względem sposobów ich realizacji, to mimo wszystko są to różne koncepcje. Lapidarnie, acz niezbyt precyzyjnie, główną różnicę między nimi można przedstawić następująco: **w stronicowaniu program i dane są mecha-**

nicznie dzielone na równe części bez zwracania uwagi na treść dzielonej informacji, zaś w przypadku segmentacji dokonuje się rozbioru logicznego informacji i dopiero na tej podstawie jest ona dzielona na części. Dużą zaletą segmentacji w porównaniu z po-



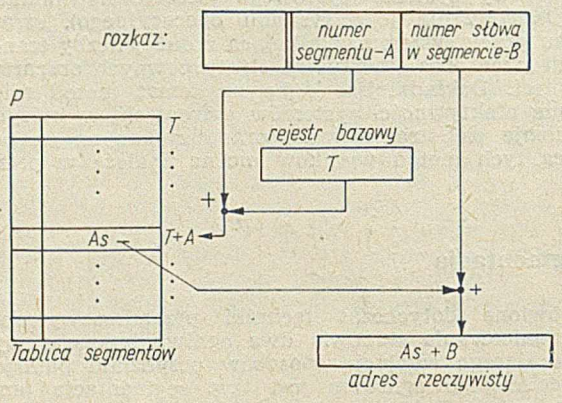
Rys. 10. Budowa rozkazu przy segmentacji ze stronicowaniem

przednimi metodami jest możliwość stosunkowo prostego zorganizowania efektywnej współpracy kilku programów z jednym podprogramem, będącym na przykład procedurą realizacji typowych i spotykanych w różnych programach czynności.

Możliwe jest połączenie koncepcji segmentowania i stronicowania, co zrobiono m.in. w systemie IBM 360/67 [16], wówczas segmenty zostają podzielone na strony, a rozkaz programu ma postać przedstawioną schematycznie na rys. 10.

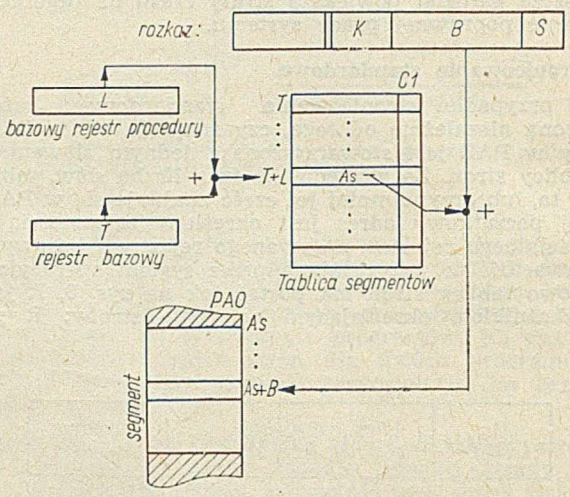
Segementacja bez stronicowania

Najprostszy sposób zapewnienia odwzorowania adresu instrukcji programu w przypadku segmentacji bez stronicowania jest analogiczny do tego, który opisano dla stronicowania. Metoda ta została schematycznie przedstawiona na rys. 11. W przypadku, gdy



Rys. 11. Sposób przekształcania adresu w segmentacji bez stronicowania

dopuszczalna liczba segmentów dla danego systemu cyfrowego jest bardzo duża, tak że podanie explicite numeru segmentu wraz z adresem wewnątrz segmentu jest niemożliwe ze względu na ograniczoną długość części adresowej instrukcji, stosuje się **bazowy rejestr procedury** oraz grupę tzw. **bazowych rejestrów adresów.** W rejestrach tych są przechowywane numery określonych segmentów. Na rys. 12 przedstawiono sposób odwzorowania adresu w przypadku, gdy przekształcony adres odnosi się do aktualnie realizowanego segmentu, tzn., gdy S = 0. Zawartością bazowego rejestru procedury jest liczba L, która po dodaniu do zawartości rejestru bazowego wskazuje właściwe słowo w tablicy segmentów (T + L), a w konsekwencji żądany adres wewnątrz segmentu (As + + B). W przypadku, gdy przekształcony adres nie odwołuje się do aktualnie realizowanego segmentu, tzn. S = 1, wykorzystuje się wskazywany przez etykietę K jeden z bazowych rejestrów adresów. W rejestrze tym przechowywany jest numer właściwego słowa w tablicy segmentów. Dalsze operacje (przedstawione schematycznie na rys. 13) są analogiczne do tych, które omówiono poprzednio. Wartość bitu S, zawartości bazowego rejestru procedury oraz bazowych rejestrów adresów są ustalane przez system operacyjny. Kilkubitowe pole C1 w słowie tablicy segmentów może służyć do przechowywania różnych informacji, np: ochrona dostępu do segmentu, typ pamięci, w której znajduje się segment (PAO lub PAZ), typ dostępu do segmentu (tylko odczyt, tylko zapis, zapis

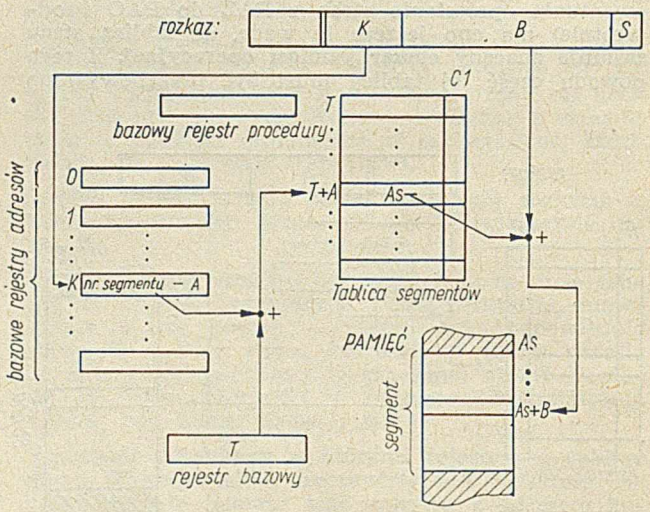


Rys. 12. Odwozrowanie adresu przy segmentacji bez stronicowania, w przypadku gdy przekształcany adres odnosi się do aktualnie wykonywanego segmentu

i odczyt) itd. Udoskonaleniem opisanego sposobu odwzorowania adresu jest wykorzystanie jednego z bazowych rejestrów adresów jako rejestru indeksującego dla innego rejestru z tej grupy. Przy segmentacji można też stosować opisaną już pamięć asocjacyjną.

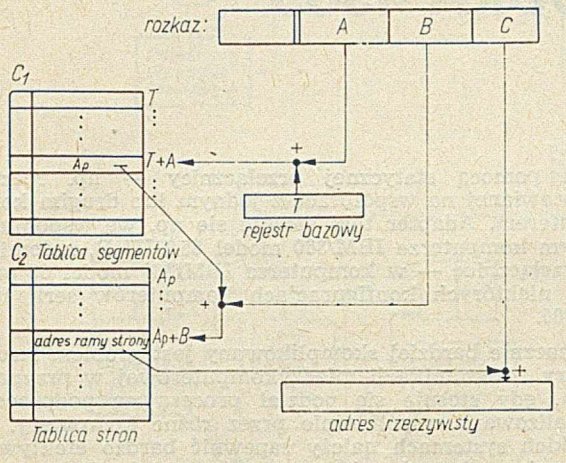
Segmentacja ze stronicowaniem

W przypadku zastosowania segmentacji ze stronicowaniem adres ma postać przedstawioną na rys. 10, tzn. składa się z trzech części: numeru segmentu A, numeru strony B i numeru komórki na stronie C.



Rys. 13. Odwzorowanie adresu przy segmentacji bez stronicowania, w przypadku gdy przekształcony adres odnosi się do aktualnie niewykonywanego segmentu.

Proces odwzorowania adresu przedstawiony schematycznie na rys. 14 jest analogiczny do omówionych poprzednio, przy czym obecnie składa się on z trzech etapów: wyznaczenie rzeczywistych adresów segmentu, strony i wreszcie — adresu żądanej komórki pamięci. Pole C2 z tablicy stron służy do przechowywania informacji podobnych do tych, które są zapisane w polu C1 w słowie z tablicy segmentów. Informacje te odnoszą się jednak do stron wewnątrz segmentu.



Rys. 14. Odwzorowanie adresu przy segmentacji ze stronicowaniem

Dokładniejszą analizę różnych sposobów segmentacji uwzględniającą problemy związane z tą metodą zawarto w pracach [15] i [4], zaś bardzo ciekawą dyskusję na temat segmentacji i stronicowania można znaleźć w listach Owena [17] i Pinkertona [18]. W dyskusji tej Owen przedstawił pogląd, że **rozwiązanie problemów przydziału pamięci w systemach cyfrowych można oprzeć na segmentacji, a stronicowanie należy traktować jedynie jako historyczny już etap wypracowania tej techniki.** Z poglądem tym ostro polemizuje Pinkerton.

Maszyna wirutalna

Zarówno stronicowanie, jak i segmentacja charakteryzują się tym, że wszystkie znajdujące się w pamięci systemu cyfrowego programy i ich dane są podzielone na części. Części te, w zależności od aktualnego stanu systemu, mogą znajdować się w PAO bądź w PAZ. Aby nie obciążać użytkownika kłopotami związanymi z koniecznością przesyłania informacji pomiędzy pamięciami obu typów, co zresztą byłoby niemożliwe również z tego powodu, iż z reguły nie bierze on udziału przy podziale swego programu na części, wykorzystano koncepcję tzw. **pamięci wirtualnej** (ang. *virtual memory*). Koncepcja ta polega na tym, że do dyspozycji użytkownika oddaje się jednolitą (w sensie programowym) pamięć o bardzo dużej pojemności, przy czym nie musi on być świadom, że **w rzeczywistości tworzą ją różne urządzenia: pamięć rdzeniowa, dyskowa, taśmowa itd.** Przesyłanie informacji z PAZ do PAO i odwrotnie, odbywa się w tak zorganizowanej pamięci bez wiedzy użytkownika.

Zarysowana w artykule organizacja pamięci systemu cyfrowego oparta na rozdzielnym lub łącznym zastosowaniu stronicowania i segmentacji zmienia w dużej mierze sposób współpracy człowieka z komputerem. Użytkownik jest obecnie postawiony „w obliczu" **maszyny wirtualnej**, tzn. takiej, która składa się z procesora, urządzeń do wprowadzania i wyprowadzania informacji oraz pamięci wirtualnej. **Autor programu nie dba o właściwe podzielenie swego programu i danych pomiędzy różne rodzaje pamięci, nie interesuje się również zorganizowaniem ewentualnych przesłań fragmentów programu czy zbioru danych pomiędzy poszczególnymi typami pamięci; wykonaniem tych czynności jest obciążony system operacyjny komputera.** Ma to ogromne znaczenie praktyczne, gdyż pozwala znacznie ułatwić proces programowania, a więc w rezultacie umożliwia rozszerzenie kręgu bezpośrednich użytkowników systemów cyfrowych.

* * *

Jak już wspomniano, rozwój metod przetwarzania pociągnął za sobą opracowanie nowych sposobów wykorzystania i organizacji pamięci systemu cyfrowego. Sposoby te, co staje się szczególnie widoczne z dzisiejszej perspektywy, również przeszły swoją ewolucję. Trzy jej ogniwa najbardziej — zdaniem autora artykułu — w tym procesie charakterystyczne zostały przedstawione w niniejszej pracy (opis innych sposobów można znaleźć np. w [4]). Splot obu tych ciągów rozwojowych zaowocował nową koncepcją komputera, a mianowicie: maszyną wirtualną. Maszyna ta zgodnie z dialektycznym prawem negacji może być traktowana — mutatis mutandis — jako **urządzenie przywracające pierwotny sposób współpracy człowieka z komputerem.** Sposób ten, jak wiadomo, polegał na tym, że użytkownik w czasie pracy maszyny cyfrowej miał ją do swej wyłącznej dyspozycji, a jedynym rodzajem pamięci jakim mógł dysponować była — mająca jednolitą strukturę — pamięć operacyjna.

Literatura

[1] Polska norma. Przetwarzanie danych i komputery. Podstawowe nazwy i określenia. PN-71/T-01016.

[2] J. Roder, A. Rosene: Memory protection in multi processing systems. IEEE TRANSC. EC-16, no. 3, June 1967.

[3] N. Weizer, G. Oppenheimer: Virtual memory management in a paging environment. AFIPS Spring Joint Comp. Conf., vol., 34, 1969.

[4] H. Dryzek: Organizacja pamięci w systemach liczących. Problemy podziału. PWN, Warszawa, 1970.

[5] J .Bertin, M. Bitout, J. Rougier: L'exploitation partagée des calculateurs Dunod, Paris, 1973.

[6] R. Daley, F. J. Dennis: Virtual memory, processes and sharing in MULTICS. COMM. ACM, vol. 11, May 1968.

[7] L. Belady: A study of replacement for a virtual-storage computer. IBM SYSTEMS JOURNAL, vol. 5, no 2, 1966.

[8] I. Flores. Virtual memory and paging. DATAMATION, August, September 1967.

[9] T. Kilburn, D. Edwards. One-level storage systems. IEEE TRANSE. EC-11, no. 2, 1962.

[10] D. Aspinall, D. Kinniment: Associative memories in large computer systems. IFIP. August 1968.

[11] N. Naturajan, P. Thomas: A multiaccess associative memory. IEEE TRANS. on Computers, May 1969.

[12] J. Demmis: Segmentation and the design of multiprogrammed computer systems. Mc Graw-Hill Book Company, 1967.

[13] P. Franaszek, T. Wagner: Some distribution? — free aspects of paging algorithm performance. JOURNAL of ACM, vol. 21, 401, 1974.

[14] D. Mitra: Some aspects of hierarchical memory systems. JOURNAL of ACM, vol. 21, no 1, 1974.

[15] J. Laski: Segmentation and virtual adress topology — an essay in virtual research. COMPUTER JOURNAL, vol. 2, May 1968.

[16] Ch. Gibson: Time-sharing in the IBM system 360 model 67. Proc. Spring Joint Comp. Conf. AFIPS, vol. 128, 1966.

[17] D. Owen: Correspondence. COMPUTER JOURNAL, vol. 11, November 1968.

[18] T. Pinkerton: Correspondence. COMPUTER JOURNAL, vol. 12, February 1969.

ANDRZEJ DZIERŻYKRAJ
WOJCIECH KUCZBORSKI

681.322

Organizacja systemów wielokomputerowych w oparciu o komputery ODRA 1305

W artykule omówiono zagadnienia związane z organizacją systemów wielokomputerowych, opartych o komputery ODRA 1305. Przedstawiono m.in. typowe konfiguracje sprzętu, ich własności operacyjne oraz instrukcje, które zapewniają elastyczne współdziałanie programów realizowanych na różnych komputerach. Omówiono także szczegółowo dwa podstawowe urządzenia zapewniające na poziomie sprzętowym współdziałanie komputerów: sterowaną przełącznicę kanałów oraz adapter międzykanałowy.

Struktury systemów wielokomputerowych

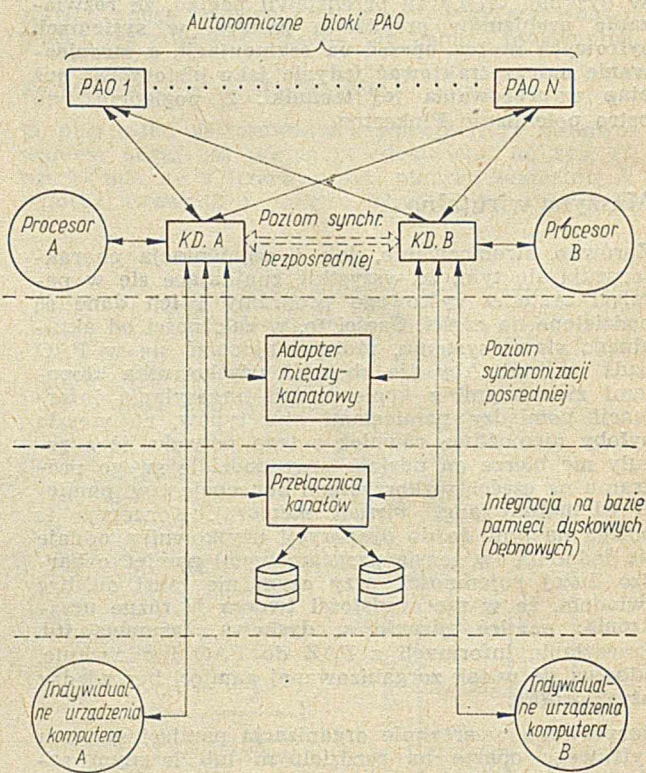
W miarę postępującej integracji przetwarzania coraz poważniejszą rolę zaczynają odgrywać systemy wielokomputerowe. Właściwie wykorzystane, pozwalają uzyskać wysoką gotowość operacyjną (zdolność do przetwarzania pomimo awarii poszczególnych elementów systemu), duży współczynnik wykorzystania środków oraz skrócenie czasu realizacji problemów o odpowiednim stopniu współbieżności. Terminem „system wielokomputerowy" oznaczamy tu system złożony co najmniej z dwu komputerów, mających możliwość bezpośredniego wzajemnego przekazywania komunikatów. Biorąc pod uwagę właściwości techniczne i programowe maszyn ODRA 1305 omówmy przypadki, w których rozwiązania wielokomputerowe są uzasadnione, lub nawet niezbędne.

Zastosowania polegające na zapewnieniu ciągłej gotowości operacyjnej systemu są od dawna znane i opisane w literaturze. Najprostsze — to użycie jednego lub dwu kompletnych komputerów rezerwowych, kiedy ich współdziałanie ogranicza się do wymiany tzw. informacji restartowej. W przypadkach bardziej wyrafinowanych, dwu lub trzykrotny nadmiar dotyczy tylko pewnych funkcji krytycznych. Pod obciążeniem znajdują się wtedy wszystkie zainstalowane komputery, a w razie awarii ulega ograniczeniu zakres przetwarzania (systemy ze stopniowym „upadkiem"). Przykładem mogą być tu dwukomputerowe zestawy IBM/360 model 65; dwukomputerowy zestaw UNIVAC 1107, sterujący ruchem kolejowym we Francji, IRS Data Entry [5], System Naziemnego Wsparcia Lotów APOLLO w Huston itp.

Takie zastosowania odciążające kosztowną i szybką jednostkę centralną, a zwłaszcza pamięć operacyjną od funkcji związanych z wolnymi urządzeniami zewnętrznymi są dobrze znane z praktyki i literatury. Należą do nich wszystkie typowe systemy z komputerami satelitarnymi do obsługi urządzeń we/wy — np. IBM/360, model 85/VIII B.

W obu powyższych przypadkach rozwiązania są stosunkowo proste, ponieważ nie wspomagają one współdziałania programów realizowanych przez różne komputery, konieczne jest jedynie wzajemne przekazywanie danych, w zasadzie nawet bez konieczności synchronizowania ich z realizowanymi programami. W tym wypadku stosuje się połączenie komputerów przez ich kanały zewnętrzne (przy użyciu adaptera międzykanałowego) i (ewentualnie) umożliwia się dostęp do wspólnie użytkowanych pamięci zewnętrznych za pomocą statycznej przełącznicy — np. ręcznie ustawianej na współpracę z jednym lub drugim komputerem. Adapter taki stosuje się np. we wspomnianym komputerze IBM/360 model 85/VIII B, natomiast przełącznicę — w komputerze IBM/360 model 65 oraz w niektórych konfiguracjach komputerów serii ICL 1900.

Znacznie bardziej skomplikowany jest problem struktury i komunikacji międzykomputerowej w przypadku, gdy stosuje się podział procesu na podprocesy realizowane współbieżnie przez różne komputery. W takich systemach należy zapewnić bardzo efektywną wymianę danych między programami i synchronizację zadań współzależnych pomiędzy komputerami. Działania te muszą być przy tym dostępne dla programisty programów użytkowych. Będą one zależne od zakresu dopuszczalnych modyfikacji standardowego oprogramowania nadzorczo-sterującego i obowiązującej dyscypliny w firmowych standardach programowych — można jednak ogólnie stwierdzić, że kom-



Rys. 1. System dwukomputerowy z różnymi poziomami integracji
KD — koordynator dostępów do PAO
PAO — pamięć operacyjna

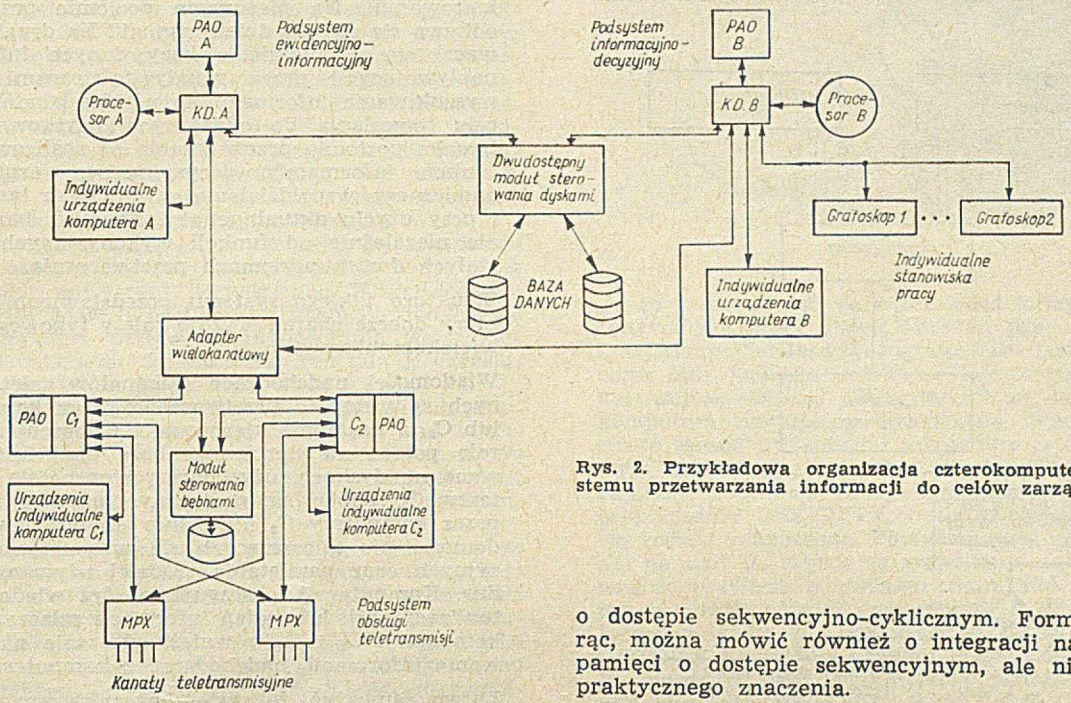
putery ODRA 1305 pozwalają zrealizować w miarę uniwersalny system wielokomputerowy przez jedynie niewielkie rozszerzenie programu nadzorczego EXECUTIVE.
System taki można rozpatrywać z trzech zasadniczych punktów widzenia: sprzętu (układy i urządzenia), ekstrakodów egzekutora (tzw. prymitywy wewnętrzne, struktury danych i procedury wykonawcze) oraz użytkownika (instrukcji). Ograniczymy się do omówienia problemów związanych ze sprzętem.

Jednym z warunków efektywności systemu jest mały procentowy udział procedur współdziałania komputerów i niewielkie obciążenie związane z wymianą komunikatów między programami. Oba te parametry można ograniczyć do dowolnie niskiego poziomu odpowiednio dobierając tzw. baze integracji systemu (rodzaj pamięci wspólnie użytkowanej przez komputery) oraz poziom synchronizacji (poziom, na którym odbywa się wymiana informacji synchronizacyjnych). Celowe jest rozróżnianie dwóch poziomów synchronizacji bezpośredniej i pośredniej. Synchronizacje bezpośrednia osiąga się wyposażając jednostki centralne w specjalne łącze międzyprocesorowe do inicjowanej za pomocą odpowiednich rozkazów wymiany informacji sterujących. Wymaga to ingerencji w logikę komputera.

Synchronizacja zewnętrzna ma miejsce wówczas, gdy wymiana informacji sterujących między komputerami odbywa się w trybie kanał—kanał, za pośrednictwem adaptera międzykanałowego.

Na poziomie programów użytkowych komputery „widzą" się wzajemnie jako zbiory, natomiast na poziomie programów nadzorczo-sterujących — jako urządzenia zewnętrzne.

Taka koncepcja pozwala zaadaptować standardowe instrukcje we/wy (w przypadku EMC ODRA 1305 rozkazy 157, 162) i przesyłać w ten sposób zarówno informacje synchronizacyjne, jak i dane do obliczeń. Jest to rozwiązanie najtańsze i najprostsze, nie wymagające żadnych adaptacji standardowego sprzętu, a jedynie zastosowania specjalnego adaptera międzykanałowego, oferowanego zresztą przez niektórych producentów — m.in. firmę ICL. Jego wadą jest niska efektywność przy realizacji programów wymagających częstej wymiany danych i informacji sterujących. Sytuacja jeszcze się pogarsza, jeżeli chodzi o dane przechowywane na stałe w pamięciach zewnętrznych. Można wyeliminować tę ostatnią trudność (uzyskując jednocześnie podwyższenie gotowości operacyjnej) dobierając odpowiednio bazę integracji systemu. Integrację stosuje się w zasadzie albo na poziomie pamięci operacyjnej albo na poziomie pamięci



Rys. 2. Przykładowa organizacja czterokomputerowego systemu przetwarzania informacji do celów zarządzania

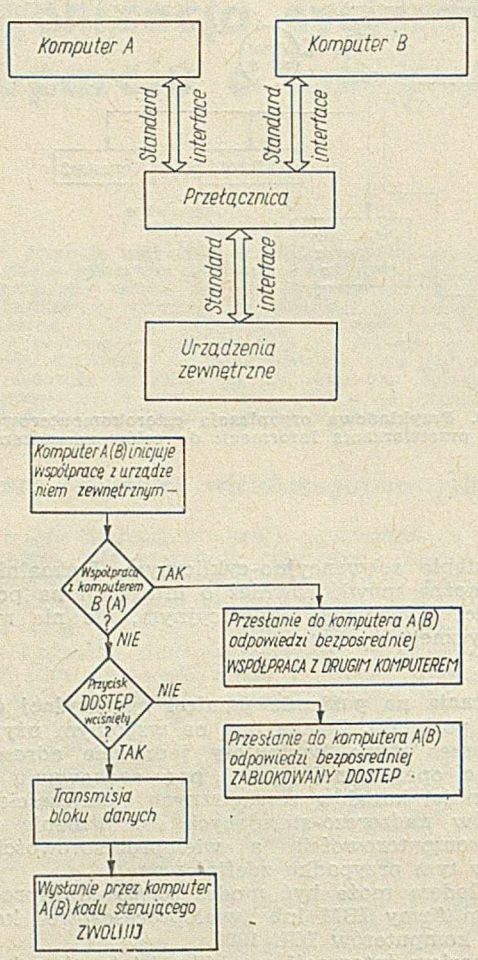
o dostępie sekwencyjno-cyklicznym. Formalnie biorąc, można mówić również o integracji na poziomie pamięci o dostępie sekwencyjnym, ale nie ma ona praktycznego znaczenia.

Integracja na poziomie pamięci operacyjnej (pierwszego poziomu). Polega ona na wspólnym użytkowaniu przez różne komputery jednolicie adresowanej pamięci operacyjnej (rys. 1), przy zachowaniu jednak odrębności kanałów zewnętrznych i najczęściej programów nadzorczo-sterujących (rozróżnienie między wielokomputerowością a wieloprocesorowością nie jest w tym przypadku zdefiniowane).
Przykładem może być model 360/65 Multiprocessing System firmy IBM lub „wieloprocesorowe" konfiguracje komputerów ICL 1900.
Ten poziom integracji ma kilka istotnych zalet:

- możliwość współbieżnego wspólnego użytkowania procedur przez różne procesory (oszczędność pamięci i zmniejszenie liczby przesyłań pomiędzy pamięcią operacyjną a pamięcią zewnętrzną)
- możliwość zapewnienia procesorom bezpośredniego dostępu do wszystkich danych znajdujących się w pamięci operacyjnej (nie ma konieczności przemieszczania danych w przypadku realizowania zadań współzależnych przez różne procesory)
- niewrażliwość systemu na awarie procesorów (o ile awarie te nie powodują odcięcia dostępu do istotnych urządzeń zewnętrznych)
- łatwo można zorganizować synchronizację zadań na poziomie programów nadzorczo-sterujących.

Omawiany typ systemu wielokomputerowego pozwala bardzo skutecznie skracać czas realizacji tych programów, które wymagają dużych obliczeń i których zadania są w dużym stopniu równoległe, a nie wymagają operowania dużymi zbiorami danych, przechowywanych w pamięciach zewnętrznych.
Utrzymanie normalnej pracy w przypadku awarii pamięci operacyjnej oraz ochrona obszarów użytkowanych przez komputery na zasadzie wyłączności przed oddziaływaniem innych komputerów, wymagają dość skomplikowanych rozwiązań układowych, których jednak nie będziemy tu omawiać. Integracja na poziomie pamięci operacyjnej nie jest możliwa bez odpowiedniego przystosowania logiki pamięci operacyjnej.

Integracja na poziomie pamięci zewnętrznych o dostępie sekwencyjno-cyklicznym (drugiego poziomu). W praktyce polega ona na wspólnym użytkowaniu pamięci bębnowych lub (i) dyskowych (rys. 1) przez różne komputery, które niezależnie od tego są najczęściej połączone ze sobą jeszcze za pomocą adaptera międzykanałowego.



Rys. 3. Schemat podłączenia urządzenia zewnętrznego do dwóch komputerów i zasada działania przełącznicy

Ten typ organizacji jest obecnie szeroko rozpowszechniony w systemach pracujących na bieżąco *(real-time)*. Typowym przykładem mogą być wspomniane już: dwumaszynowa konfiguracja UNIVAC 1107 (sterująca ruchem kolejowym we Francji) lub IRS Data Entry firmy GE.

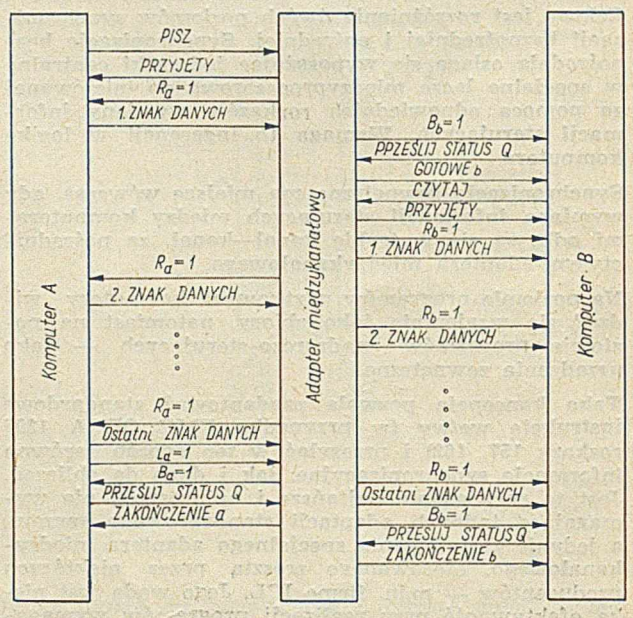
Dwudostępność pamięci zewnętrznych musi w tym przypadku mieć charakter dynamiczny — dostęp może w każdej chwili uzyskać każdy z podłączonych komputerów pod warunkiem, że urządzenie nie jest właśnie zaangażowane we współpracę z innym komputerem. Wymaga to jednak albo specjalnie zaprojektowanych urządzeń sterujących albo zastosowania specjalnych przełącznic przystawnych. Te ostatnie mają tę zaletę, że pozwalają wykorzystać standardowe wyposażenie i organizować system wielokomputerowy w oparciu o istniejący system tradycyjny. Dodatkową zaletą wielodostępności pamięci zewnętrznych jest możliwość znacznego zredukowania liczby rezerwowych urządzeń pamięciowych, jak i zmniejszenie liczby dodatkowych przesyłań w zastosowaniach, w których wielokomputerowość wprowadza się w celu uzyskania wysokiego poziomu gotowości operacyjnej. W tym przypadku wielodostępność stosuje się także dla magnetycznych pamięci taśmowych (czasem innych droższych urządzeń, jak np. drukarki wierszowe).

Interesujące możliwości efektywnego wykorzystania integracji drugiego poziomu występują w systemach przetwarzania w czasie rzeczywistym o wyraźnej wielopoziomowości procesu przetwarzania (np. systemy kierowania). Na pierwszym poziomie przetwarzania odbywa się obsługa teletransmisji. Na drugim — prowadzi się uaktualnienie bazy danych informacjami napływającymi przez kanały teletransmisyjne oraz wyszukiwanie informacji do użytku lokalnego lub w celu przesłania do oddalonych użytkowników. Na trzecim poziomie przetwarzania są realizowane różne funkcje informacyjno-decyzyjne, w warunkach dynamicznego współdziałania człowieka z komputerami i przy użyciu aktualnych informacji z bazy danych, ale niezależnie od funkcji wykonywanych na pozostałych dwóch poziomach przetwarzania.

Strukturę takiego systemu przedstawiono na rys. 2, który dobrze charakteryzuje zalety typowego systemu wielokomputerowego.

Wiadomości nadchodzące z kanałów teletransmisyjnych są wstępnie przetwarzane przez komputery C_1 lub C_2, a następnie kierowane do komputera A. Jego rola polega na sterowaniu bazą danych, zorganizowaną na dyskach, użytkowanych wspólnie z komputerem B. Bęben magnetyczny, używany wspólnie przez komputery C_2 i C_1 służy do buforowania wiadomości nie mieszczących się w pamięciach operacyjnych oraz pamiętania rzadziej używanych procedur. Komputer A obsługuje, oprócz wiadomości aktualizacyjnych lub żądań informacji zgłaszanych przez komputery C_1 i C_2, również zgłoszenia na wyszukiwanie informacji, nadchodzące z komputera B.

Łatwo zauważyć, że system charakteryzuje się podwyższoną niezawodnością, ponieważ C_1 i C_2 tworzą symetryczny podsystem dwukomputerowy obsługi teletransmisji, a komputer B może w razie potrzeby przejąć funkcje aktualizacyjno-informacyjne związane z bazą danych. System ten można zbudować przy użyciu standardowego sprzętu, pod warunkiem posiadania adapterów międzykanałowych i sterowanej przełącznicy kanałów dla urządzenia sterującego bębnami i dyskami. Urządzenia takie dla komputerów ODRA 1305 omawiamy w następnych punktach.



Rys. 4. Zasada transmisji danych zainicjowanej przez komputer A

Sterowana przełącznica kanałów

Zadaniem przełącznicy jest umożliwienie współpracy dowolnego urządzenia zewnętrznego (np. pamięci dyskowej, multipleksora, drukarki) z dwoma komputerami serii ODRA 1300. Przełącznica podłączona jest do kanałów dowolnego typu, a jedynym wymaganiem dla urządzenia zewnętrznego jest zgodność ze standardem interfejsu.

W przeciwieństwie do ręcznie sterowanego przełącznika PSI — 304 [3], [4], stan przełącznicy zmieniają programy obu komputerów, które w sposób dynamiczny dzielą między sobą czas korzystania z urządzenia. Przełącznica przydziela urządzenie zewnętrzne temu komputerowi, który wcześniej podjął inicjatywę transmisji. Po przesłaniu bloku danych komputer zwalnia urządzenie, wysyłając kod sterujący ZWOLNIJ (kod sterujący jest to rozkaz wysyłany przez jednostkę centralną do urządzenia).

Jeżeli komputer zgłasza się do zajętego urządzenia zewnętrznego, to informuje o tym generowana przez przełącznicę odpowiedź bezpośrednia WSPÓŁPRACA Z DRUGIM KOMPUTEREM, a transmisja danych nie jest przerywana (odpowiedź bezpośrednia jest to informacja wysyłana do jednostki centralnej, określająca możliwość realizacji kodu sterującego).

Zgłaszający się komputer może w wyjątkowych przypadkach przerwać transmisję wysyłając kod sterujący PRZERWIJ WSPÓŁPRACĘ. Warunkiem przydzielenia urządzenia jest wcześniejsze wciśnięcie przez operatora stabilnego przycisku DOSTĘP (każdy komputer ma oddzielny przycisk).

Zmiana stanu przełącznicy powoduje wysyłanie do odpowiedniego komputera sygnału przerwania (linia interfejsu B = 1). Po otrzymaniu sygnału przerwania komputer bada jego przyczynę wysyłając kod sterujący PRZEŚLIJ STATUS i otrzymuje w odpowiedzi aktualny stan przełącznicy, tzw. status, dzięki czemu jest na bieżąco informowany o możliwościach współpracy z urządzeniem zewnętrznym.

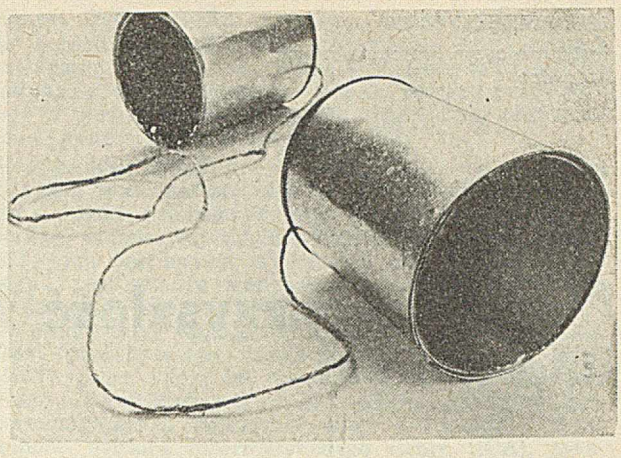
Oczywiście dynamiczna zasada pracy przełącznicy wymaga kłopotliwych zmian w programach obsługi urządzeń zewnętrznych. Dlatego przewidziano możliwość ręcznego, statycznego przełączania urządzenia, nie wymagającego jakichkolwiek modyfikacji tych programów.

Układ logiczny przełącznicy zajmuje jedynie 3 płytki drukowane (po 30 kostek małej integracji). Dlatego możliwe jest jej zainstalowanie w jednostce sterującej większości urządzeń zewnętrznych.

Niewątpliwie doskonalszym, bardziej uniwersalnym rozwiązaniem byłaby przełącznica umożliwiająca jednoczesne wykorzystanie przez dwa komputery mechanizmów podłączonych do wspólnej jednostki sterującej. Pozwoliłoby to np. na zainicjowanie przez komputer A przesuwu głowic magnetycznych w wybranym mechanizmie pamięci dyskowej. Na czas tego przesuwu pozostałe mechanizmy byłyby dostępne dla obu komputerów. Ponieważ czas ten jest stosunkowo długi (max 150 ms dla pamięci dyskowej ODRY 1305), pozwoliłoby to na lepsze wykorzystanie kosztownego urządzenia. Oczywiście konieczne jest dostosowanie jednostki sterującej do tego rodzaju pracy.

Adapter międzykanałowy

Zadaniem adaptera jest pośredniczenie w wymianie danych między dwoma komputerami serii ODRA 1300, realizowanej za pośrednictwem kanałów. Oczywiście część przesyłanych danych może pełnić funkcję rozkazów zlecanych przez jeden komputer, a wykonywanych przez drugi. Adapter, pracujący zgodnie z zasadami standardowego interfejsu „widziany" jest przez obie jednostki centralne jako typowe urządzenie wejścia-wyjścia. Dane wysyłane są zawsze od komputera inicjującego transmisję. Rysunek 4 ilustruje zasadę transmisji z komputera A do B. Pierwszym krokiem jest przesłanie do adaptera kodu sterującego PISZ, na który wysyłana jest odpowiedź bezpośrednia PRZYJĘTY, o ile poprzednia transmisja została zakończona. Pojawienie się jedynki logicznej na linii R_a interfejsu sygnalizuje gotowość adaptera do przyjęcia pierwszego znaku, który przesyłany jest w chwilę później z komputera A do rejestru buforowego adaptera. Powoduje to ustawienie statusu GOTOWE i wysłanie sygnału przerwania B_b. Z kolei komputer B bada przyczynę przerwania wysyłając kod sterujący PRZEŚLIJ STATUS Q, stwierdza, że istnieje status GOTOWE, wysyła kod sterujący CZYTAJ. Po ustawieniu $R_b = 1$ następuje odebranie pierwszego znaku przez komputer B.



Kolejne znaki bloku danych przesyłane są dzięki pojawianiu się w odpowiednich momentach sygnałów na liniach R_a i R_b. Zapalenie linii L_a w trakcie przesyłania ostatniego znaku powoduje pojawienie się statusów ZAKOŃCZENIE$_a$ i ZAKOŃCZENIE$_b$, a tym samym ustawienie B_a i B_b. Transmisję kończy sprawdzenie przez oba komputery przyczyny przerwania. Transmisja w kierunku przeciwnym przebiega analogicznie. Wykrycie przez adapter lub komputer odbierający dane błędu parzystości sygnalizowane jest przez jednoczesne pojawienie się statusów GOTOWE$_a$, ZAKOŃCZENIE$_a$, GOTOWE$_b$, ZAKOŃCZENIE$_b$. Również adapter zajmuje 3 płytki po 30 kostek i może być wbudowany do jednej z jednostek centralnych.

Literatura

[1] Inter — program communication, „User Notice" do podręcznika Central Processors ICL 1904 (4095).

[2] Standard interface ODRA/1300 — Norma Branżowa 1973 r.

[3] J. Piasecki: Przełącznik Standard Interface typ PSI-304. INFORMATYKA 1974, 10.

[4] K. Koleśnik: Przełącznik Standard Interface PSI 304. INFORMATYKA 1973, 12.

[5] IRS Data Entry Systemy, DATAMATION, 1970, June.

JERZY MIŁEK
Instytut Maszyn Matematycznych
Katowice

681.3.002.2(44)

Niepewna przyszłość francuskiego przemysłu informatyki

Artykuł omawia w aspekcie gospodarczym i politycznym sytuację francuskiego przemysłu informatyki.

Ubiegły rok był dla francuskiego sektora przemysłu i usług informatycznych okresem niezbyt pomyślnym. Ogólna, kryzysowa sytuacja gospodarcza rozwiniętych krajów kapitalistycznych odbiła się także na rozwoju przemysłu środków elektronicznego przetwarzania danych. Kurczące się z miesiąca na miesiąc portfele zamówień oraz znaczny spadek zapotrzebowania na usługi informatyczne, to główne czynniki, które spowodowały znaczne zaostrzenie walki głównych producentów o rynki światowe. W walce tej stroną najbardziej aktywną są oczywiście wielkie firmy Stanów Zjednoczonych, dysponujące relatywnie ogromnymi zasobami finansowymi, bardziej zdywersyfikowaną strukturą produkcji oraz większym doświadczeniem konstrukcyjnym i marketingowym.
Jedną z najpoważniejszych przyczyn zaostrzenia się konkurencji ze strony takich kompanii jak CONTROL DATA CORPORATION, HONEYWELL czy XEROX są wzrastające kłopoty „największego z największych" — IBM — którego seria procesów z amerykańskimi organami antytrustowymi stwarza doskonałą okazję do przeprowadzenia nowego podziału światowych rynków informatycznych.
Pierwszym celem amerykańskich kompanii sektora informatycznego stała się informatyka francuska. Pierwszą, zresztą nieudaną, próbą były pertraktacje CONTROL DATA w sprawie przejęcia kontroli nad największą francuską firmą software'ową SIA (Société d'Informatique Appliquée) w kwietniu 1974 roku.

Niekonsekwencje „Planu Calcul"

Od uchwalenia w roku 1966 rządowego „Planu Calcul", którego podstawowym celem było stworzenie narodowego, konkurencyjnego przemysłu informatycznego, największą trudnością w realizacji tego ambitnego zamierzenia była potrzeba skoordynowania ogólnogospodarczych preferencji wyrażonych w programie rządowym z dążeniami prywatnego sektora przemysłu, któremu powierzono zadanie umocnienia przemysłowej bazy „Planu Calcul".
„Plan Calcul" w dziedzinie przemysłu bazował na daleko idącej koncentracji i umocnieniu pozycji konkurencyjnej czysto francuskich producentów środków informatyki, przy czym operacją najpoważniejszą było połączenie dwóch firm wytwarzających jednostki centralne — Société Europeenne d'Automatisme oraz Compagnie Europeenne d'Automatisme Électronique — i stworzenie w wyniku tej fuzji kompanii CII (Compagnie Internationale pour l'Informatique). Rząd francuski nie zdecydował się w 1966 roku na przejęcie bezpośredniej kontroli nad „pupilkiem Francuzów", ograniczając swe oddziaływanie do zapewnienia znacznej pomocy finansowej w formie dotowania spodziewanego deficytu nowej firmy i jej działalności badawczo-rozwojowej, skupienia w CII większości zamówień administracji państwowej na sprzęt informatyczny oraz mniejszościowego udziału w kapitale zakładowym (za pośrednictwem rządowej agencji rozwoju IDI w wysokości 23% akcji).
Kontrolę finansową nad CII powierzono spółce holdingowej FININFOR, której głównymi udziałowcami są THOMSON-BRANDT oraz COMPAGNIE GÉNÉRALE D'ELECTRICITÉ. Te dwie grupy zawsze konkurowały między sobą w różnych dziedzinach przemysłu elektrotechnicznego, elektronicznego, a zwłaszcza w produkcji środków łączności. W tych warunkach mariaż obu konkurentów w łonie FININFOR nie wyszedł, jak się później okazało, na dobre nowo utworzonej firmie.
Stałe, zakulisowe przetargi o kontrolę nad sposobem wykorzystania znacznych subwencji rządowych na prace badawczo-rozwojowe (w okresie lat 1966—1971 wynosiły one 580 mln franków, w latach 1971—1974 wzrosły one do 800 mln) połączone były z marginalną rolą sektora informatycznego w ogólnej strukturze produkcji obu partnerów. Prowadzenie prestiżowej działalności w zakresie produkcji maszyn matematycznych kosztowało bardzo drogo. Dopóki władze państwowe, za pośrednictwem Delegatury ds. Informatyki, pokrywały rosnący deficyt CII, zatargi pomiędzy głównymi akcjonariuszami kompanii nie wywoływały poważniejszych reperkusji.
Sytuacja zmieniła się diametralnie wraz z ogłoszeniem przez nowego prezydenta Francji — Valerego Giscard d'Estainga — szerokiego programu reorganizacji rządowych instytucji nadzorujących działalność poszczególnych dziedzin przemysłu. Jednym z najbardziej zaskakujących posunięć wynikających z programu było zlikwidowanie Delegatury ds. Informatyki w połowie sierpnia ubiegłego roku, co powszechnie przyjęte zostało jako wyraz rządowej dezaprobaty dla dotychczasowych postępów w realizacji „Planu Calcul", a zwłaszcza społecznych kosztów tworzenia narodowego przemysłu informatycznego.
W połowie ubiegłego roku upłynął termin ustalenia wysokości nowych dotacji rządowych dla CII na lata 1975—1976. Parokrotne przesuwanie terminu podjęcia ostatecznej decyzji w tej sprawie pogłębiło jeszcze bardziej trudności kompanii. Klienci CII rekrutujący się z prywatnego sektora gospodarki krajowej, o pozyskanie których trwała kilkuletnia akcja, przesuwają w czasie realizację swoich zamówień lub otwarcie rezygnują z zakupów w firmie, której przyszłość nie jest określoną.

JERZY MIŁEK ukończył studia w Wyższej Szkole Ekonomicznej w Katowicach w 1971 r. Obecnie pracuje w Śląskim Oddziale Instytutu Maszyn Matematycznych, w Pracowni Analiz Międzynarodowych.

W tej sytuacji różnice zdań na temat dalszej działalności CII ujawniły się z całą ostrością. Dyrekcja kompanii CGE zainteresowana jest w zachowaniu „status quo" z uwagi na możliwość zacieśnienia współpracy CII z CIT-ALCATEL (filią CGE) w zakresie systemów telekomunikacji na potrzeby Poczty Francuskiej. Natomiast kierownictwo THOMSON-CSF otwarcie opowiada się za bezpośrednim udziałem instytucji rządowych w zarządzaniu CII. Popierało ono projekt przejęcia znacznej części udziałów w kapitale zakładowym CII przez rządowy Komisariat do spraw Energii Atomowej — CEA.

Projekty „łagodnej nacjonalizacji" CII nie znalazły jednakże uznania w kołach rządowych i gospodarczych. Wydaje się, że głównym czynnikiem, który odstręcza władze od przejęcia bezpośredniej kontroli jest obawa o dalsze kontynuowanie działalności zalążka zjednoczonego przemysłu informatycznego Zachodniej Europy, jakim jest UNIDATA — grupująca poza CII także SIEMENSA i PHILIPSA.

W rok po utworzeniu UNIDATY, wprowadziła ona pierwsze komputery pod wspólną marką (seria 7700), zbliżone konstrukcyjnie najbardziej do rodziny maszyn francuskich typu IRIS. Za wcześnie jeszcze na ocenę handlowego powodzenia nowej serii, lecz wiadomo już, że znaczne koszty jej przygotowania obarczyły w najpoważniejszym stopniu stronę niemiecką, co wynika między innymi z małego doświadczenia SIEMENSA w samodzielnej produkcji maszyn matematycznych. Ubiegłoroczny deficyt sektora informatycznego SIEMENSA, oceniany na 200 mln marek, przyczynił się do pogłębienia w Niemczech niesprzyjającej dla kontynuowania eksperymentu europejskiego atmosfery.

Najbardziej zainteresowaną w powodzeniu UNIDATY jest, jak się wydaje, strona francuska, lecz w obecnych warunkach oznacza to konieczność przejęcia przez nią w ugrupowaniu roli wiodącej, zarówno na płaszczyźnie technologicznej, jak i finansowej. Dlatego też CII w wyniku reorganizacji — a taka uważana jest obecnie za konieczną — stać się musi firmą silniejszą organizacyjnie i finansowo. Według powszechnej opinii panującej we francuskich kołach branżowych, warunek ten nie może zostać spełniony w wyniku przejęcia kontroli nad CII przez jakąkolwiek instytucję państwową. Żadna z nich nie chce dobrowolnie przyjąć na siebie odpowiedzialności za powodzenie ugrupowania na szczeblu europejskim — jednoznaczne stanowisko w tej sprawie Komisariatu ds. Energii Atomowej jest tego dobitnym przykładem.

Alternatywa amerykańska

Rozwiązaniem, które obecnie wydaje się najbardziej prawdopodobnym, jest organizacyjne i kapitałowe powiązanie „niechcianego dziecka" z jedną z firm amerykańskich.

Głównym partnerem jest aktualnie HONEYWELL-BULL, jedyny „europejski" producent maszyn matematycznych, który w ubiegłym roku osiągnął w tej dziedzinie nadwyżkę bilansową. Jak dotąd, w oficjalnych wypowiedziach kierownictwa kompanii macierzystej HONEYWELL przeważa niezdecydowanie co do ewentualnych korzyści umocnienia swej pozycji we Francji. Oceniać to można wyłącznie jako wybieg taktyczny w celu zdobycia lepszej pozycji przetargowej w pertraktacjach z rządem francuskim oraz grupami kontrolującymi dotychczas CII.

Stawka jest ogromna — bezpośrednia lub pośrednia kontrola nad znaczną częścią europejskiego przemysłu środków informatyki oraz perspektywa poważnej pomocy finansowej rządów krajów zainteresowanych. Głównym problemem wydaje się być wysokość „posagu", jaki władze francuskie gotowe są przydzielić ewentualnej spółce HONELWELL-CII.

Do alternatywy współpracy z producentem amerykańskim przychyla się także strona niemiecka. SIEMENS przez wiele lat współpracował z przemysłem Stanów Zjednoczonych (w zakresie produkcji maszyn matematycznych skorzystał wiele z likwidacji działalności sektora informatycznego firmy RCA), natomiast PHILIPS, będący w łonie UNIDATY stroną najbardziej bierną, z pewnością zaaprobuje każdą próbę rokującą nadzieje na zmniejszenie deficytu kosztownego ugrupowania.

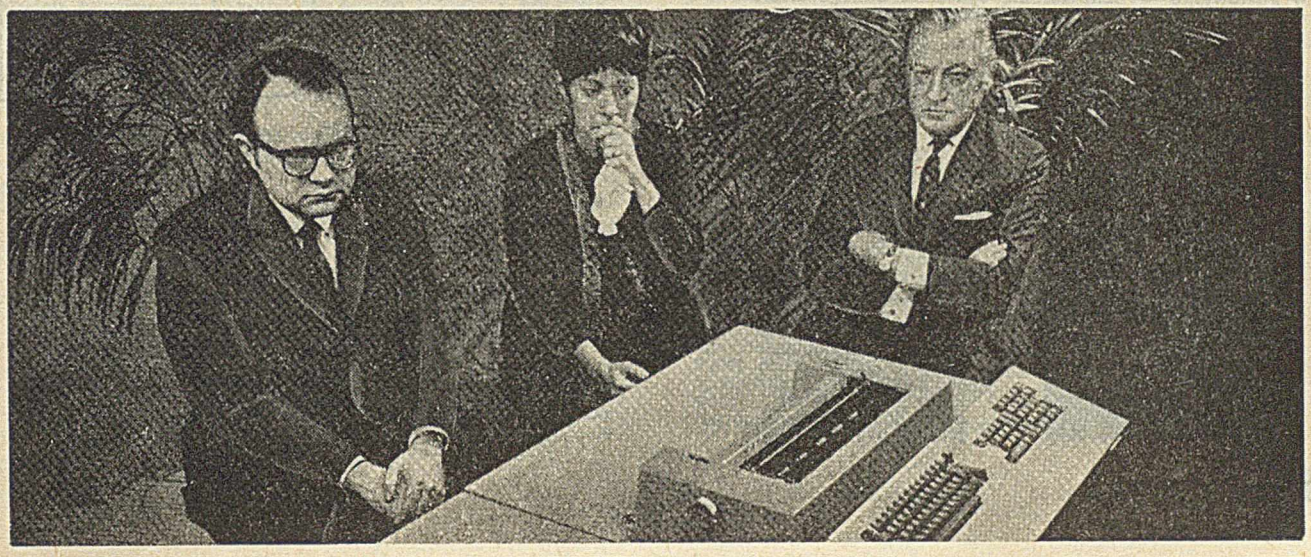
Niezależnie od ostatecznego wyniku prowadzonych negocjacji, stwierdzić należy, iż droga do stworzenia niezależnego, silnego przemysłu środków informatyki w obrębie EWG jest niezmiernie trudna. Producenci zachodnioeuropejscy jeszcze raz okazali się niezdolni do prowadzenia skutecznej walki o rynek z konkurentami zza oceanu.

W obecnych warunkach dla CII najważniejszym problemem jest przyspieszenie momentu podjęcia ostatecznej decyzji w sprawie organizacyjnego statusu firmy, co jest jedyną szansą zmniejszenia rujnującej niepewności w zakresie perspektyw rozwojowych.

Domaga się tego kierownictwo przedsiębiorstwa, a zwłaszcza sami pracownicy CII. Trudności wynikające z poważnego zmniejszenia się portfelu zamówień spowodowały już pierwsze redukcje personelu w największych zakładach CII w Tuluzie. Znamienną jest postawa związków zawodowych, które traktując przedłużające się przetargi jako „powolne niszczenie" przedsiębiorstwa, zainicjowały masową akcję protestacyjną wśród załogi.

Instancje związkowe przypominają wyniki „afery" BULL z roku 1964, kiedy to rząd francuski i koła finansowe nie były zdolne do zrealizowania swego hasła „obrony narodowych interesów", oddając kontrolę nad największym francuskim producentem maszyn matematycznych w ręce GENERAL ELECTRIC. Jak wiadomo, pociągnęło to za sobą znaczne redukcje załogi BULL w wyniku przeprowadzenia przez koncern amerykański „Programu usprawnienia produkcji i organizacji".

Reasumując, w aspekcie finansowym, rok 1975 stać się może momentem przełomowym dla rynku informatycznego krajów kapitalistycznych. Procesy wytoczone IBM, XEROX, trudności finansowe ICL oraz stagnacja w całym przemyśle europejskim mogą spowodować głębokie zmiany w podziale rynków zachodnich.



JĘDRZEJ WRÓBLEWSKI
Centrum INTE
Warszawa

681.322.004.14:681.616.7

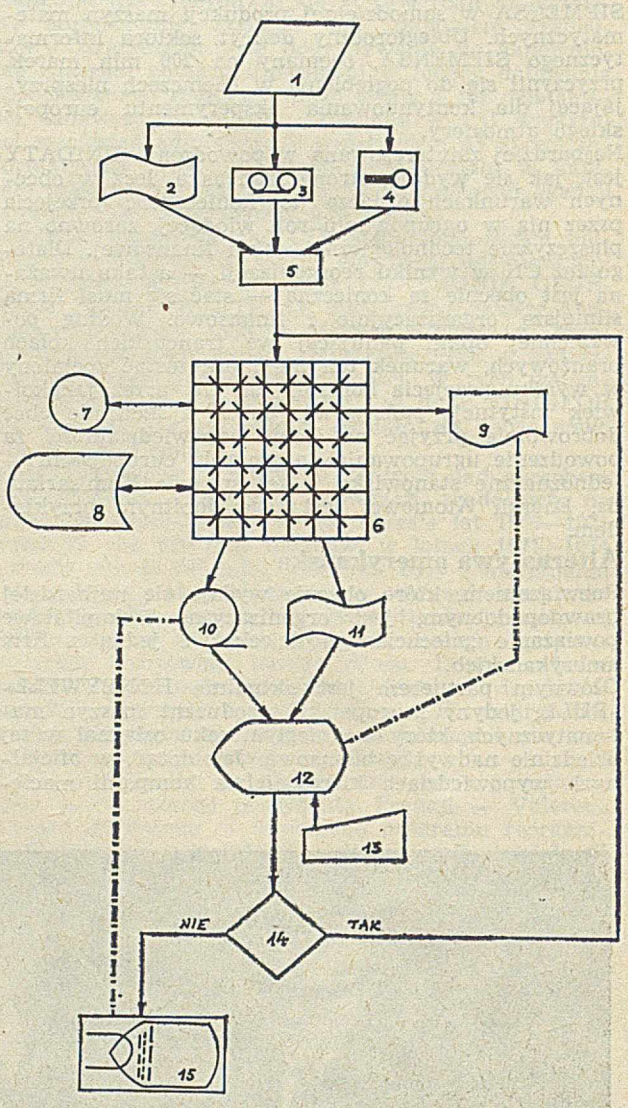
Komputeryzacja prac edytorsko-poligraficznych. Fotoskład

Artykuł omawia stan komputeryzacji prac edytorsko-poligraficznych ze szczególnym uwzględnieniem fotoskładu, przedstawiając istniejący obecnie sprzęt, oprogramowanie i zastosowania.

Lata sześćdziesiąte charakteryzowały światowy przemysł poligraficzny dynamiką rozwoju szybkiego składu fotograficznego, popularnie nazwanego fotoskładem, który w pracach przygotowawczych przy druku offsetowym (płaskim) nabiera coraz większego znaczenia. W technologii fotoskładu tekst składany jest bezpośrednio na materiale światłoczułym, najczęściej na błonie fotograficznej, a nie odlewany całymi wierszami lub pojedynczymi z czcionkami z gorącego, płynnego metalu. W technice druku maszynami offsetowymi zaczęto stosować fotoskład przy składaniu tytułów, akcydensów, tabel, wzorów chemicznych i matematycznych. Teksty naświetlone na materiale światłoczułym charakteryzują się dobrą ostrością i doskonałym zaczernieniem, dając jako produkt końcowy gotowy do montażu diapozytyw.

Schematy technologiczne komputeryzacji fotoskładu

Z dużej ilości możliwych do zrealizowania systemów automatyzacji procesów fotograficznego składania tekstu można wybrać kilka podstawowych, odpowiadających najczęściej spotykanym w praktyce [1]. Jako konfiguracje optymalne długoletnia praktyka zaakceptowała układy składające się z aparatów kodujących bez justowania [1]) (stacje przygotowania danych na nośnikach maszynowych), monitorów ekranowych (stacje korekty) i automatu do fotoskładu sprzężonego z minikomputerem. Schemat takiego wieloprzebiegowego podsystemu z korekcją przedstawiono na rysunku 1.
Wejście do podsystemu stanowi taśma papierowa lub magnetyczna zawierająca niewyjustowany tekst oraz informacje kontroli formatu poprzedzone specyfikacją zadania [2]. Specyfikacja tą ustala parametry, które nie zmieniają się w czasie wykonywania zadania, np. odstęp pomiędzy wierszami. Podczas pierwszego przebiegu w komputerze, tablice opisujące



Rys. 1 — Wieloprzebiegowy podsystem składu fotograficznego tekstu z korektą.
1 — manuskrypt (oryginał)
2 — dziurkarka klawiaturowa (zakodowany tekst)
3 — kasetowa taśma magnetyczna (zakodowany tekst)
4 — dysk elastyczny (zakodowany tekst)
5 — specyfikacja zadania, niewyjustowany tekst
6 — komputer
7 — magnetyczna taśma z tablicą wydruków

[1]) Justowanie — uzupełnienie spacjami danego wiersza do, określonej wcześniej, jego szerokości.

Mgr inż. JĘDRZEJ WRÓBLEWSKI ukończył indywidualne studia poligraficzne Politechniki Warszawskiej 1973. Specjalizował się w elektronicznych metodach korekcji barw i składzie fotograficznym sterowanym komputerem. Pracuje w Centrum INTE w Zakładzie Koordynacji Systemów. Autor kilku artykułów i opracowań popularno-naukowych.

8 — dyski magnetyczne ze słownikiem wyjątków
9 — wydruk kontrolny na drukarce wierszowej
10 — komputerowa taśma magnetyczna (tekst przetworzony)
11 — taśma papierowa (tekst przetworzony)
12 — monitor ekranowy
13 — klawiatura monitora ekranowego
14 — decyzja — czy są błędy?
15 — automat do fotoskładu

żądaną postać wydruku pobierane są z dostarczonej przez producenta taśmy bibliotecznej programów [2], a przetworzony tekst zapisywany jest na komputerowej taśmie magnetycznej. Jednocześnie na drukarkę wierszową wyprowadzany jest wydruk kontrolny. Na podstawie analizy tego wydruku przygotowuje się na nośniku wejściowym poprawki wykrytych błędów, które oprócz zmian w tekście mogą zawierać modyfikację specyfikacji zadania i ewentualne skorygowanie rozkazów kontroli formatu. Zmodyfikowany na taśmie magnetycznej tekst jest ponownie wyprowadzany w postaci wydruku kontrolnego na drukarce wierszowej. Procedura wykrywania i korygowania błędów powtarzana jest do momentu otrzymania na komputerowej taśmie magnetycznej ostatecznej postaci tekstu wraz z rozkazami sterującymi automatem do fotoskładu.

Automaty do składu fotograficznego przystosowane są do pracy zarówno w trybie *off-line*, jak i *on-line* w oparciu o cztery kody podstawowe:

— ISO *(International Standard Organisation)*
— ECMA *(European Computer Manufacturers Association)*
— EBCDIC *(Extended Binary Coded Decimal Interchange Code)*
— ASCII *(American Standard Code for Information Interchange).*

Wyżej omówiony podsystem nie wykorzystuje wszystkich możliwości, jakie wynikają z zastosowania komputera w technologii fotoskładu, dlatego należy traktować go jako schemat podstawowy, umożliwiający tworzenie systemów o bardziej złożonych funkcjach.

Jeden z nich, zrealizowany w oparciu o podsystem poprzednio omówiony, opiera się na koncepcji banku danych [3] przedstawiono na rys. 2. Jest to konfiguracja urządzeń pozwalająca na wprowadzenie do pamięci komputera informacji pochodzących z ośrodków przetwarzania danych tekstowych, pracujących na różnych nośnikach informacji. Zróżnicowanie nośnika informacji jest zjawiskiem powszednim i wynika z faktu dużej rozbieżności czasowej tworzenia poszczególnych ośrodków przetwarzania tekstu. Magazynowanie informacji opiera się na 1/2-calowych taśmach magnetycznych lub dyskach magnetycznych.

Stworzenie tak rozbudowanego systemu wynika z charakteru działalności tego typu ośrodka świadczącego w znacznym zakresie usługi dla drukarń, które nie posiadają własnego komputera. W systemach tych wykorzystuje się wyłącznie uniwersalne maszyny cyfrowe, np. IBM 360, SIEMENS 4004. dysponujące obszerym i bardziej wszechstronnym oprogramowaniem.

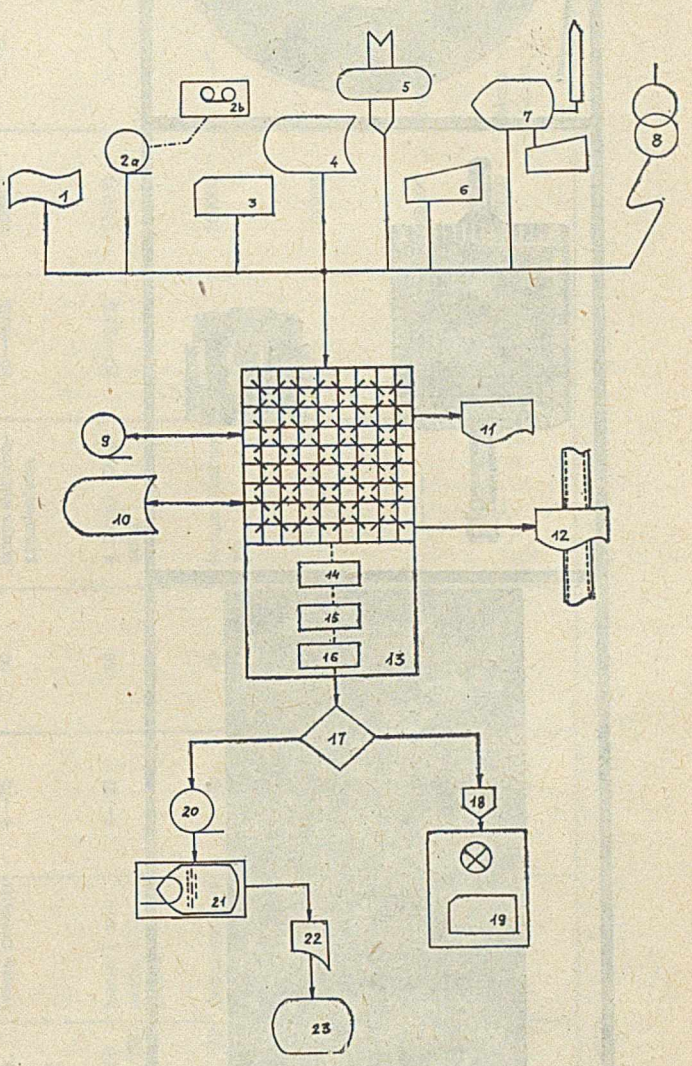
Automatyzacja składu jest tu tylko częścią całego systemu przetwarzania, obejmującego korekty, sortowania, indeksowania, łamania kolumn itp. Uniwersalne maszyny cyfrowe zapewniają ponadto lepszą ekonomikę pracy w porównaniu do komputerów specjalizowanych, realizujących jedynie operacje justowania.

Niektóre spośród specjalizowanych maszyn cyfrowych np. GSA Computer System, oprócz programu typograficznego, wykonują program automatycznego dzielenia wyrazów. Jednakże maszyny te nie mogą realizować innych zadań, niż bezpośrednio związanych ze składem fotograficznym.

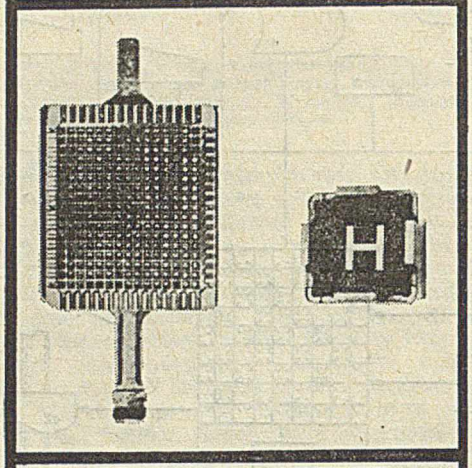
[2] W czasie odwoływania się do tych tablic, tekst jest składany w wiersze, justowany i przenoszony (dzielenie wyrazów).
[3] Systemy działające w oparciu o bank danych spotkać można obecnie na świecie wyłącznie w dużych, najczęściej usługowych, wyspecjalizowanych centralnych ośrodkach fotograficznego składania tekstu.

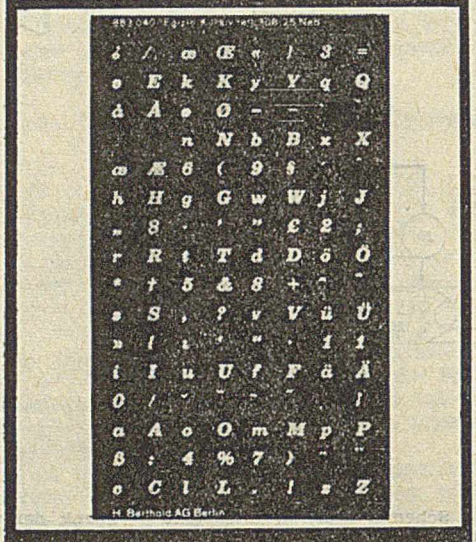
Oprogramowanie

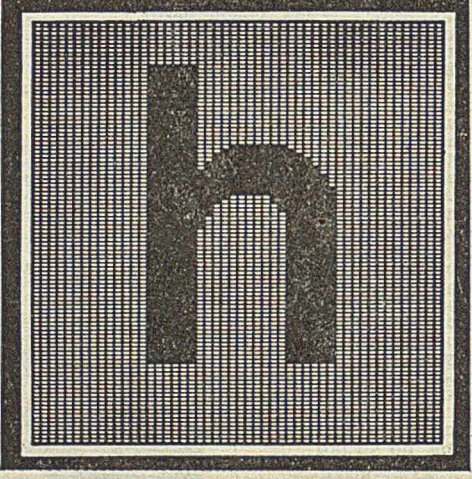
Programy związane z przetwarzaniem tekstów mogą być pisane w różnych uniwersalnych językach programowania. Najbardziej efektywnym do tych zastosowań językiem jest — jak wykazała praktyka — ASSEMBLER i dlatego wszystkie programy specjalistyczne dostarczane przez producentów komputerów napisane są w tym języku. Podstawowy program standardowy uwzględnia wszystkie parametry typograficzne, tzn.:
— format składu
— krój pisma



Rys. 2 — Schemat systemu fotoskładu — „bank danych"
1 — taśma papierowa
2a — komputerowa taśma magnetyczna
2b — kasetowa taśma magnetyczna
3 — karta dziurkowana
4 — dysk magnetyczny
5 — optyczny czytnik tekstu OCR (Optical Character Recognition)
6 — klawiatura o dostępie swobodnym
7 — monitor ekranowy z klawiaturą i piórem świetlnym
8 — modem
9 — jednostki taśm magnetycznych (tablice wydruków, słownik, wyjątki)
10 — jednostki pamięci dyskowych (bilioteka podprogramów)
11 — szybka drukarka wierszowa
12 — drukarka mikrofilmowa (Computer Output Microfilm)
13 — pamięć operacyjna
14 — zespół justowania
15 — zespół obłamywania
16 — zespół przetwarzania składu
17 — decyzje
18 — wybór nośnika
19 — technologia mikrofiszowania
20 — taśma magnetyczna (tekst przetworzony)
21 — automat do fotoskładu
22 — forma do druku
23 — szpalty i strony fotoskładu







Rys. 3 — Przegląd nośników znaków
3a — dysk znakowy
3b — ramka z wymiennymi matrycami
3c — ramka z niewymiennymi matrycami
3d — litera z nośnika digitalnego

— wielkość odstępu międzywierszowego
— stopień pisma
— ukształtowanie wiersza (środkowanie, justowanie do lewego lub (i) prawego marginesu)
— odmiana kroju pisma (normalne, półgrube, grube, pochyłe)
— odmiana charakteru pisma (rozciągnięte, ściągłe)
— obłamywanie (format, liczba wierszy)
— spacjowanie
— wcięcie akapitowe
— składanie w „chorągiewkę"
— liczba wierszy w kolumnie
— liczba łamów w kolumnie
— zmiany położenia podstawowej linii pisma.

Program ten również automatycznie określa i wpisuje w właściwym miejscu numerację stron.

Za pomocą komputera można także automatycznie otrzymać spis treści, skorowidze, numery stron w odsyłaczach, wpisywać wszelkiego rodzaju przypisy i żywe paginy.

Rozbudowując programy edytorskie należy jednak pamiętać, że nadmierne rozszerzenie wymagań stawianych tym programom zmusza do stosowania maszyn cyfrowych o szczególnie dużej pojemności pamięci operacyjnej.

Nośniki znaków

Zaletą automatów do fotoskładu, rozpatrywanych jako urządzenia wyjściowe komputera, jest szczególnie szeroki asortyment znaków pisarskich. Pod pojęciem asortymentu należy rozumieć nie tylko liczbę znaków, lecz również gamę stopni pism, odmiany ich krojów i odmiany charakteru tych krojów. Liczba możliwych do uzyskania znaków w różnych modelach automatów waha się od kilku do kilkudziesięciu tysięcy. A więc — w porównaniu do konwencjonalnych drukarek komputerowych (maksymalna możliwość do 240 różnych znaków) stwarza olbrzymie możliwości wydawnicze w problemach przetwarzania informacji tekstowych. Typowe automaty pozwalają na wyprowadzanie znaków o wymiarach od 4 do 26 lub 32 punktów typograficznych [4], a niektóre z nich mają rozszerzony zakres do 72, a nawet 112 punktów, co zapewnia uzyskanie znaków o znacznych rozmiarach niezbędnych do tekstów tytułowych, np. w przypadku automatycznego składania dzienników.

Nośnikami znaków w automatach są między innymi dyski znakowe (rys. 3a), ramki z wymiennymi (rys. 3b) i niewymiennymi fotomatrycami (rys. 3c), bębny znakowe, listwy literowe lub tzw. niematerialne nośniki znaków, popularnie nazywane digitalnymi (rys. 3d). Digitalne nośniki znaków — to kombinacje binarne znaków pisarskich zakodowane w pamięci ferrytowej. Wydobycie żądanego znaku odbywa się przez podanie jego adresu w pamięci. Do tego celu wykorzystywane są pamięci typu ROM *(Real Only Memory)*.

Rozwiązanie wykorzystujące matryce niewymienne należy traktować jedynie jako magazyn uzupełniający dyski lub bębny znakowe, gdyż używanie ich obniża znacznie wydajność urządzenia naświetlającego. Najczęściej spotykanymi nośnikami w automatach do fotoskładu, o wydajności kilkudziesięciu tysięcy znaków na godzinę, są dyski znakowe.

Systemy składu fotograficznego

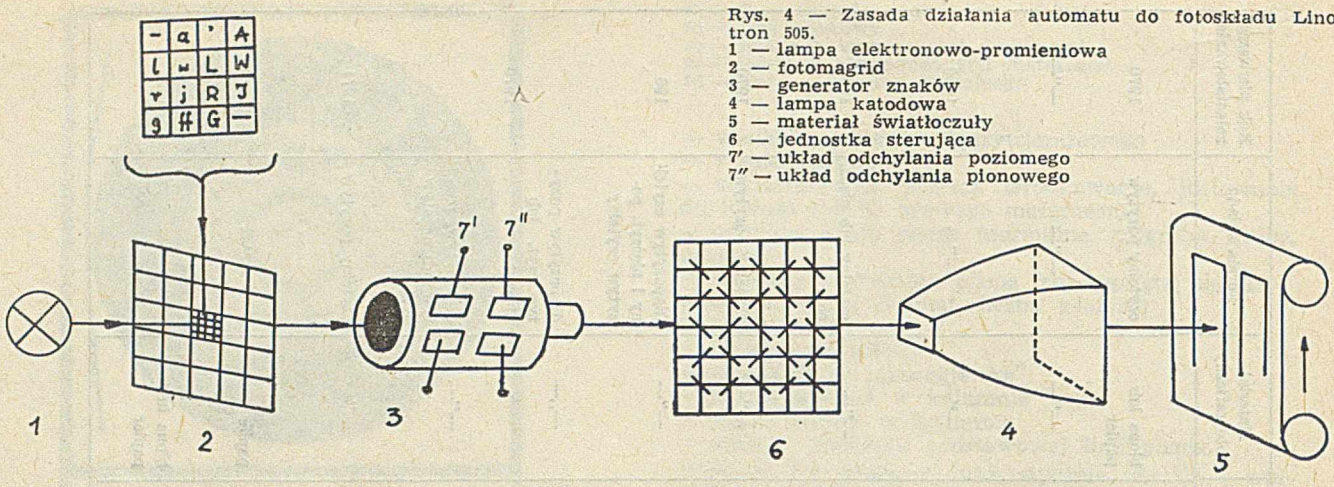
Nowoczesne automaty do fotoskładu wykorzystują w charakterze nośnika znaków strumień elektronów, który pobudza do luminescencji luminofor lampy

[4] Punkt typograficzny — elementarna jednostka miary w poligrafii; wynosi 0,376 mm.

Tabela, Dane techniczne niektórych automatów do fotoskładu*)

Automat	Producent	Konstrukcja	Sposób sterowania	Nośnik znaku	Wielkość pisma w pkt. typograficznych	Max. długość wiersza w cycerach	Źródła światła	Realizacja układowa	Wydajność (zn/h)	Materiał fotograficzny	Forma składu	Rok pierwszego zainstalowania
CAT	Graphic Systems (USA)	Jednostka fotograf.	8-ścieżkowa TP**	Dyski znakowe	5—18	44	lampa Xenonowa	MSI	90000	błona lub papier	czytelny pozytyw	1970
Fototronic TxT	Harris — Intertype Corp. USA	—"—	6—8 ścieżkowa TP	—"—	5—36	39	promień plazmy	IC	270000	—"—	—"—	—"—
Fototronic CRT	—"—	Jedn. sterująca i naświetlająca	7—9 ścieżkowa TM***	digitalny	4—144	94	lampa promieniowo—elektron	IC	3600000	—"—	—"—	1968
Digiset 50 T21	Dr inż. R.Hell (RFN)	—"—	8 ścieżk. TP i 9 ścieżk. TM lub on-line	—"—	4—32	167	—"—	IC—TTL	do 7,2 mln	—"—	czatelny lub niecz. poz.	1969
Linofilm-Europa	Linotype GmbH (RFN)	jednostka fotografująca	on-line	bęben znakow	6—24	40	lampa Xenonowa	IC	40000	—"—	czytelny pozytyw	1970
Linotron 505 c	Linotype-Paul (W. Bryt.)	jedn. sterująca i naświetl.	6—8 ścieżk. TP i 7—9 ścieżk. TM	tarcze znakowe	4—72	60	lampa elektron-promieniowa	IC—RTL	260000	—"—	diapozytyw czytelny i nieczyt. pozytyw czytelny	196
Monophoto 600	Monotype Corp (W. Bryt.)	jednostka fotografująca	8-ścieżk. TP i 9 ścieżk. TM	tarcze i magazyn fotogridów	6—28	50	4 lampy Xenonowe	IC—TTL	126000	—„—	diapozytyw pozytyw czyt. lub nieczyt.	1970
Compstar	Star Parts Co. (USA)	—"—	6,7,8 ścież. TP	bęben znakowy	5—48	45	lampa Xenonowa	MSI i LSI	180000	—"—	diapozytyw, pozytyw lub negatyw czyt.	
Photonix	Singer Business Machines (USA)	—"—	6—ścieżk. TP	tarcza znakowa	6—36	47	—"—	IC	60000	—"—	czytelny pozytyw	1971
CG 2970	Compugraphic Corp. (USA)	—"—	7-ścieżk. TP	fotomatryce	5—12	31	—"—	—	50000	papier	—"—	1970
Pacesetter	Photon Inc. (USA)	—"—	6.7,8 ścieżk. TP	dysk znakowy	5—72	54	—	IC—RTL	100000	błona lub papier	—"—	1971

*) — obecnie produkowanych jest około 75 różnych modeli
**) TP — taśma papierowa
***) TM— taśma magnetyczna



Rys. 4 — Zasada działania automatu do fotoskładu Linotron 505.
1 — lampa elektronowo-promieniowa
2 — fotomagrid
3 — generator znaków
4 — lampa katodowa
5 — materiał światłoczuły
6 — jednostka sterująca
7′ — układ odchylania poziomego
7″ — układ odchylania pionowego

oscyloskopowej (CRT). Jest to tzw. trzecia generacja maszyn do składu fotograficznego. Poprzedzająca ją generacja (druga) posługiwała się dyskiem znakowym lub fotomatrycami (urządzenia typu Photon — Lumitype lub Linofolm). Pierwszą generację cechowało natomiast używanie zwykłych negatywów fotograficznych oraz kamer (urządzenia typu Fotosetter lub Monophoto).

Typowym i najbardziej znanym w naszym kraju automatem trzeciej generacji jest Linotron 505, sterowany komputerem HONEYWELL 316, którego zasadę działania (rys. 4) podajemy niżej.

Papierowa taśma, analizowana przez czytnik fotoelektryczny wywołuje sygnały, które ośmioma światłowodami przesyłane są do układów impulsowo-logicznych realizujących określone funkcje logiczne (3). Wszystkie układy logiczne zbudowano z elementów scalonych TTL realizujących podstawowe elementy kombinacyjne typu NOR (negacja sumy).

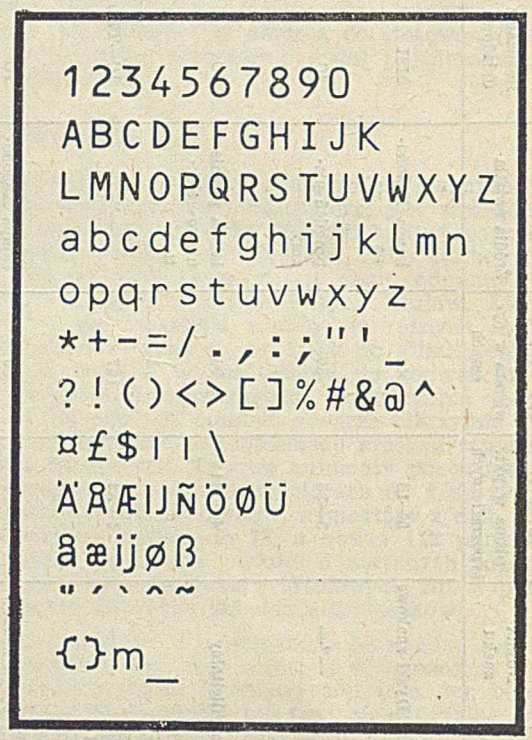
Źródło światła, jakiego użyto, to klasyczna lampa elektronowo-promieniowa, która naświetla błonę fotograficzną, dając jako produkt ostateczny — czytelny diapozytyw. Na fotomatrycę (2) o pojemności 2^8 lub 2^9 znaków pada strumień świetlny (1), który rzutowany jest na lampę elektronową generatora znaków (3), mającego za zadanie wybór właściwego znaku. Sygnał odpowiadający określonemu znakowi zostaje przesłany do lampy katodowej (4). Analogicznie odbywa się przesłanie kolejnych znaków — aż do momentu, kiedy zostanie wyprowadzona cała kolumna, która następnie w całości naświetlona jest na materiał światłoczuły (5), zwinięty w rolę. Informacja o znaku, który powinien być wygenerowany przez generator, pochodzi od jednostki sterującej (6). Mówiąc inaczej, określony znak pojawiający się w generatorze zależy od sygnałów z jednostki sterującej, które programowo wpływają na zmianę potencjału okładek kondensatorów układu odchylania poziomego (7′) i pionowego (7″), co równoznaczne jest z wyborem żądanego znaku — mającego stałe przyporządkowanie w układzie kartezjańskim x—y. Linotron 505 sprzężony jest z automatyczną wywoływaczką materiału światłoczułego tak, że na wyjściu pojawia się wywołany, wysuszony, gotowy do montażu produkt.

Wszystkie systemy nowoczesnych automatów do fotoskładu są w zasadach działania bardzo zbliżone do wyżej omówionego. Różnice wynikają jedynie z zastosowanego nośnika znaków, systemu sterowania i wyprowadzania znaku.

W tabeli I zaprezentowano przegląd automatów najczęściej stosowanych.

Komputery

Komputery używane do współpracy z urządzeniami do fotoskładu trzeciej generacji są komputerami uniwersalnymi z odpowiednio rozbudowanym oprogramowaniem wynikającym ze specyfiki prac edytorsko-poligraficznych. Oprogramowanie to stanowi wynik wieloletnich prac naukowo-badawczych takich czołowych producentów komputerów, jak SIEMENS, DIGITAL EQUIPMENT, IBM, ICL, DATA GENERAL oraz HONEYWELL. Najbardziej typowymi przedstawicielami komputerów przystosowanych do sterowania automatami do składania światłem są: SIEMENS 404, SIEMENS 4004/35 i 4004/127, PDE 8/L i 8/I, IBM 1130, ICL 1901 A, NOVA 1200, H 316 i H 312.



Rys. 5 — Krój pisma OCR-B2.

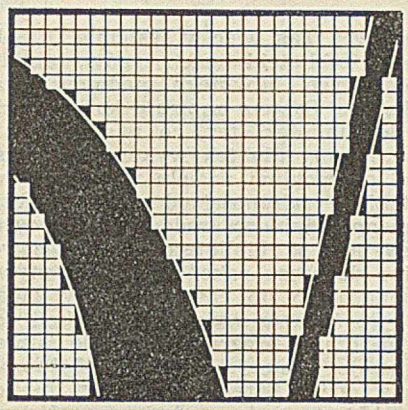
Urządzenia wejścia/wyjścia

Urządzenia we/wy komputerów stosowanych w ośrodkach składu fotograficznego pracują na typowych nośnikach informacji, tzn. taśmach papierowych, kartach perforowanych i taśmach magnetycznych. W chwili obecnej obserwuje się wyraźną tendencję do odchodzenia od kart dziurkowanych jako nośnika szczególnie niedostosowanego do wymagań prac edytorskich, pomimo ich zalety, jaką jest dogodność dokonywania korekt. Światowe trendy rozwojowe w tej dziedzinie wysuwają obecnie na czoło stosowanie kasetowych taśm magnetycznych oraz dysków elastycznych (*floppy disc*).

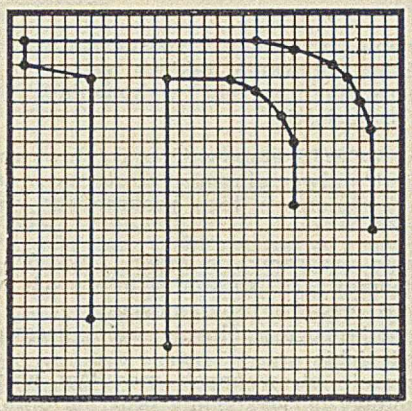
Urządzeniom do przygotowania danych stosowanym w technologii fotoskładu stawiane są wyższe wymagania typograficzne w porównaniu do innych zastosowań EPD, co niejednokrotnie stanowi poważny problem. Urządzenia te muszą mieć bowiem znacznie bogatszy repertuar znaków (duże i małe litery, znaki diakrytyczne i specjalne, pismo proste i pochyłe itp.), w związku z czym zaistniała konieczność rozbudowania ich klawiatury. Charakteryzują się one aż poczwórnym obłożeniem rejestrów tzn. operator ma do swojej dyspozycji rejestr dolny i górny lewy oraz dolny i górny prawy.

Analogiczna sytuacja występuje w przypadku drukarek wierszowych, na których wyprowadzany jest półprodukt kompilacji, monitorów ekranowych do przeprowadzenia korekty oraz optycznych czytników tekstu [5]. Obecnie produkowane optyczne czytniki tekstu, przystosowane do czytania pisma OCR-B2 (rys. 5) nie uwzględniają polskich znaków diakrytycznych [6].

Rys. 6 — Porównanie systemów generowania znaków alfanumerycznych



6a — efekt formy zaokrąglonej w systemie CRT



6b — litera tworzona w układzie wektorowym przez ruchomy punkt świetlny.

Kierunki rozwoju fotoskładu

Jednym z głównych problemów przy odtwarzaniu alfabetu nie zaprojektowanego specjalnie dla fotoskładu metodą CRT, są wszystkie linie, które do urządzenia odtwarzającego nie biegną dokładnie poziomo lub pionowo, lecz tworzą formy zaokrąglone, które określa się jako *drop-outs* i *pick-ups* (rys. 6a). Chąc ten niepożądany efekt wyeliminować, w USA prowadzone są próby nad zastąpieniem znaków digitalnych znakami tworzonymi przez ruchomy punkt świetlny [4]. Za pomocą urządzenia CRT typu Stromberg-Carlson 4020 Microfilm Printer wykreślono alfabet w układzie linii wektorowych układu współrzędnych składającym się z 2^{10} lub 2^{12} pól (rys. 6b).

[5]) Patrz: J. L. Kulikowski — „Kierunki rozwojowe i metody automatycznej klasyfikacji" — Informatyka 4 i 5/1974.

Każda litera kreślona jest z pewnej liczby odcinków prostych o kierunkach niekoniecznie zgodnych z kierunkiem linii współrzędnych przy zachowaniu zasady, że końce i początki elementarnych wektorów pokrywają się z węzłami siatki. Ruch punktu świetlnego tworzącego literę sterowany jest cyfrowo według właściwego algorytmu dla danej litery. Sygnałem dla wykreślenia najmniejszego elementu są współrzędne wektora przesunięcia punktu świetlnego, w przyjętym układzie węzłów.

Obecnie trwają bardzo interesujące prace, prowadzone głównie w USA (Battelle Memorial Institute — Columbus Laboratories) wykorzystania promieni laserowych w urządzeniach do składu fotograficznego. Tak więc, w niedalekiej już przyszłości powstanie niewątpliwie czwarta generacja fotoskładu oparta na spójnym świetle monochromatycznym. Eksperci nie wykluczają możliwości stworzenia generacji pośredniej, będącej kombinacją lamp CRT z laserem. Urządzenia te będą pozwalały na składanie z szybkością do 100 000 znaków na sekundę, tzn. 36 mln znaków składu na godzinę.

Ulegnie również zmianie digitalny nośnik znaków. Już w chwili obecnej zakończono uwieńczone sukcesem prace nad konstrukcją nowego nośnika. Zakłady Precision Instrumento w Palo Alto wyprodukowały, w oparciu o systemy laserowe i nową technikę odblaskową z zastosowaniem zmiennego pola magnetycznego, nośnik, w którym na powierzchni wielkości główki zapałki umieszczono 130 000 różnych znaków alfanumerycznych.

Bibliografia

[1] J. Maier: Verfahrenswege des Fotosatzes — „Offsetpraxis" 3/74.

[2] Computer Composition System — System ODRA 1300 — Wrocławskie Zakłady Elektroniczne ELWRO — Wrocław 1970.

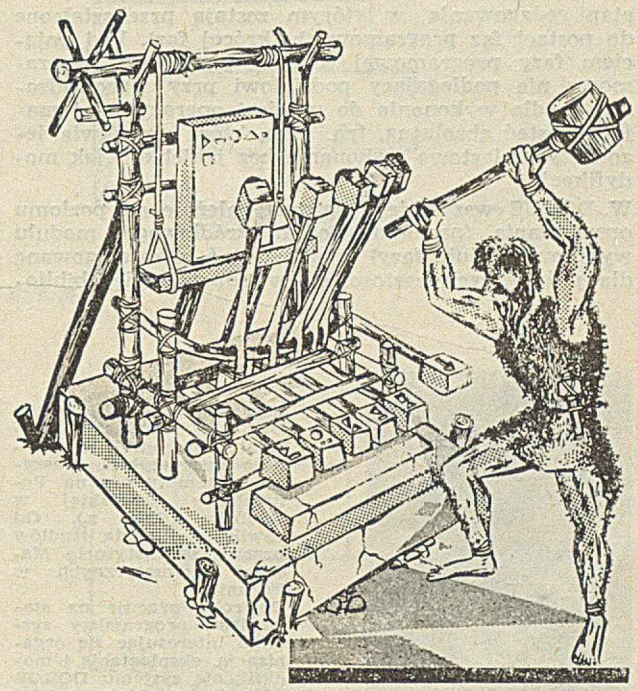
[3] J. Wróblewski: Nowoczesna drukarnia gazetowa „Anderston Quay" — POLIGRAFIKA 12/1973.

[4] H. Zapf: Koniec klasycznej typografii a nadchodzący czas liczb progresywnych — LITERA 46—4/1971.

[5] Composition Information Service, Inc — Analysis of Phototypesetting Systems — Los Angeles, California — 1972.

[6] Deutscher Drucker — rocznik 1973

[7] The seybold report, PHOTOTYPESETTERS — USA, 1973.



[6]) Polskie znaki diakrytyczne opracowane w Ośrodku Pism Drukarskich przy OBR Przemysłu Poligraficznego.

LECH JAGODZIŃSKI
Instytut Maszyn Matematycznych
Toruń

681.322.06

Struktury nakładkowe programów w systemie DOS/JS

Artykuł omawia istotę, technikę i zastosowania struktur nakładkowych i podaje przykład tworzenia struktury nakładkowej fragmentu konkretnego programu dla DOS/JS.

Przed omówieniem zasadniczego tematu niniejszego artykułu należy bliżej rozpatrzyć pewne związane z nim aspekty systemu operacyjnego DOS/JS.
System operacyjny DOS/JS udostępnia programiście języki programowania dostosowane do różnych klas problemowych. Do nich należą FORTRAN, RPG i PL/1. Języki te nie są przewidziane dla konkretnych maszyn i dlatego w niewielkim stopniu uwzględniają ich specyfikę. Inaczej jest z językiem ASSEMBLER — językiem maszynowo zorientowanym. Wyższość tego języka polega głównie na tym, że pozwala na pełne wykorzystanie możliwości maszyny. W systemie DOS/JS program napisany w dowolnym z wymienionych powyżej języków programowania polega w procesie translacji przekształceniu z postaci źródłowej na postać modułu wynikowego, który jest modułem programowym w języku pośrednim, wspólnym dla wszystkich translatorów systemu. Specyfika źródłowego języka programowania jest w ten sposób traconą. Moduły wynikowe nie są jednak jeszcze programami maszynowymi. Muszą przejść one etap redagowania, w którym zostają przekształcone do postaci faz programowych (krócej faz). Pod pojęciem fazy programowej rozumie się obiekt programowy nie podlegający podziałowi przy wprowadzaniu go dla wykonania do pamięci operacyjnej i mający postać absolutną, tzn. taką, która umożliwia jego natychmiastowe wykonanie bez jakichkolwiek modyfikacji.
W DOS/JS wszystkie programy zależnie od poziomu opracowania (poziomu modułu źródłowego, modułu wynikowego lub fazy) mogą zostać zakatalogowane dla późniejszych zastosowań w odpowiedniej bibliotece systemowej. Tak więc w systemie DOS/JS istnieją systemowe biblioteki trzech typów: biblioteka modułów źródłowych, biblioteka modułów wynikowych i biblioteka faz. System zapisuje każdy zredagowany program do biblioteki faz, co wiąże się z tym, że wszelkie programy dla ich wykonania przesyłane są do pamięci operacyjnej w celu wykonania właśnie z tej biblioteki. W zależności od zadanego reżimu pracy systemu operacyjnego zredagowany program przechowywany jest w bibliotece faz czasowo (opcja LINK), albo do czasu zakończenia wykonywania danego zadania, albo do chwili, gdy w zadaniu tym wystąpi nowe żądanie wykonania funkcji redagowania. Po tym czasie program przestaje istnieć w systemie w swojej postaci absolutnej. Może zostać jednak zadana opcja katalogowania do biblioteki faz (opcja CATAL) stanowiąca odmienny reżim pracy systemu operacyjnego. Polega on na tym, że zredagowany program przechowywany jest w bibliotece faz na stałe.

Katalogowanie programu w postaci absolutnej daje możliwość wywoływania takiego programu do pamięci operacyjnej w celu wykonania w dowolnej chwili w przyszłości.

Przyjęta w DOS/JS etapowość przygotowywania programów jest uniwersalnym i elastycznym środkiem w rękach świadomego tego programisty umożliwiającym mu:

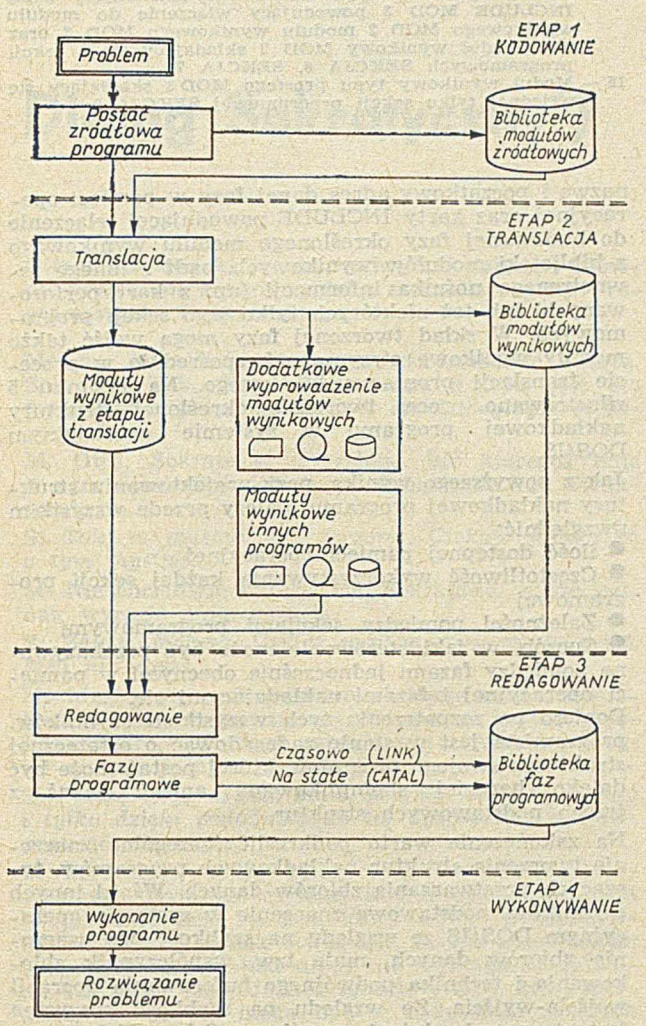
— rozbijanie programu na części i wybór dla nich najodpowiedniejszych spośród istniejących w systemie DOS/JS języków programowania
— tłumaczenie poszczególnych części programu niezależnie od siebie
— redagowanie otrzymanych modułów wynikowych w jeden gotowy do wykonania program o najbardziej efektywnej strukturze.

Moduł wynikowy może być rozpatrywany przez program redagujący (LNKEDT) jako niepodzielna jednostka przetwarzania lub jako zbiór bardziej drobnych części nazywanych sekcjami programowymi. Sekcje programowe należące do jednego modułu wynikowego są redagowane niezależnie i na tym etapie mogą być umieszczone w różnych fazach programowych. Podziału modułu wynikowego na sekcje programowe programista dokonuje już na poziomie postaci źródłowej jego programu. Programista może też włączać do swego programu gotowe moduły z biblioteki modułów wynikowych.

Wywoływanie wszystkich faz zredagowanego programu z biblioteki faz realizuje podprogram SUPERVISORA (SUPERVISOR — program nadzorczy w systemie operacyjnym DOS/JS). Wywoływanie przeprowadza się w wyniku specjalnych żądań. Żądanie wywołania pierwszej fazy programu użytkownika wydaje program sterowania zadaniami (JOB CONTROL

Mgr inż. LECH JAGODZIŃSKI ukończył Wydział Elektroniki, specjalność automatyka i maszyny matematyczne na Politechnice Gdańskiej w Gdańsku (1972 r.). Od chwili ukończenia studiów pracuje w Instytucie Maszyn Matematycznych w Toruniu.
Od roku pracuje na stanowisku programisty systemu interesując się organizacją, eksploatacją i modyfikacją systemu DOS/JS Jednocześnie bierze udział w pracach nad oprogramowaniem systemów EPD dla JS EMC.

— program należący do systemu operacyjnego DOS/ /JS) na podstawie karty sterującej typu EXEC. Żądania wywołań pozostałych faz tegoż programu wydają sam program problemowy. Istnieją dwa sposoby wywoływania faz — poprzez makrorozkaz FETCH



Rys. 1

lub poprzez makrorozkaz LOAD. W wyniku makrorozkazu FETCH podprogram SUPERVISORA wywołuje z biblioteki faz tę, której nazwa została wskazana w makrorozkazie i umieszcza ją w pamięci operacyjnej poczynają od adresu zdefiniowanego na etapie jej redagowania. Następnie SPERVISOR przekazuje sterowanie wywołanej fazie. Wywołanie fazy poprzez makrorozkaz LOAD stosuje się wówczas, gdy nie trzeba przekazać sterowania wywołanej fazie natychmiast po jej wywołaniu. Podprogram SUPERVISORA wywołuje z biblioteki fazę o nazwie wymienionej w makrorozkazie LOAD i umieszcza ją w pamięci operacyjnej pod adresem, który może zostać zadany w tymże makrorozkazie (adres ten nie musi się pokrywać z adresem zdefiniowanym na etapie redagowania tej fazy, o ile jest ona typu samoprzemieszczalnego, tzn. jest zbudowana w ten sposób, że może się wykonywać w dowolnym miejscu pamięci operacyjnej). Po wywołaniu fazy sterowanie wraca do punktu programu bezpośrednio za makrorozkazem LOAD.

Opisany schemat pracy systemu operacyjnego DOS/ /JS ilustruje rysunek 1.

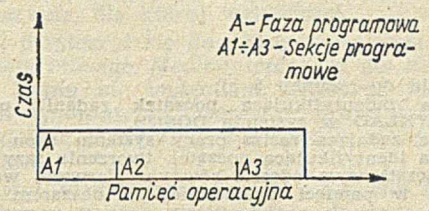
Struktury nakładkowe programów

Tworzenie struktury nakładkowej programu jest techniką programowania, która w istotny sposób minimalizuje objętość wymaganej przez program pamięci operacyjnej.

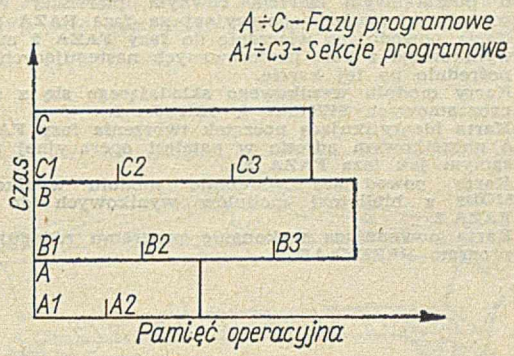
Z modułów wynikowych, jak już powiedziano, program redagujący tworzy fazy programowe składające się na dany program. W systemie operacyjnym DOS/JS fazy programowe mogą tworzyć struktury trzech podstawowych typów: struktura prosta, struktura nakładkowa z fazą podstawową oraz struktura nakładkowa bez fazy podstawowej.

- Struktura prosta zakłada, że program składa się z jednej fazy programowej w całości wywoływanej do pamięci operacyjnej w celu wykonania (rysunek 2).
- Strukturą nakładkowa z fazą podstawową zakłada istnienie fazy podstawowej, tzn. fazy, która w całości lub częściowo znajduje się w pamięci operacyjnej przez cały czas wykonywania programu. Faza podstawowa wywoływuje do pamięci operacyjnej pozostałe fazy nakładając je jedna na drugą (rysunek 3)
- Struktura nakładkowa bez fazy podstawowej zakłada, że program składa się z faz, z których każda po wykonaniu wywołuje następną fazę do pamięci operacyjnej na swoje miejsce (rysunek 4).

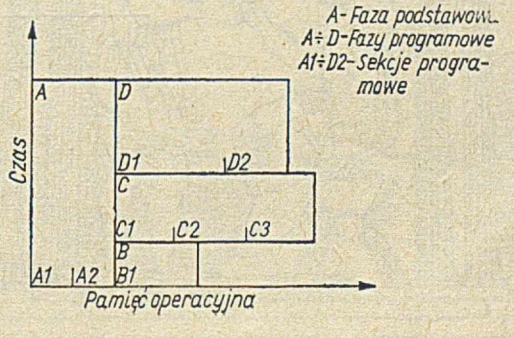
Rysunki 2, 3 i 4 są tylko graficzną reprezentacją, ilustrującą w jaki sposób fazy mogą wykorzystywać pamięć operacyjną w różnym czasie. Nie określają one jednak kolejności wykonywania poszczególnych faz.



Rys. 2

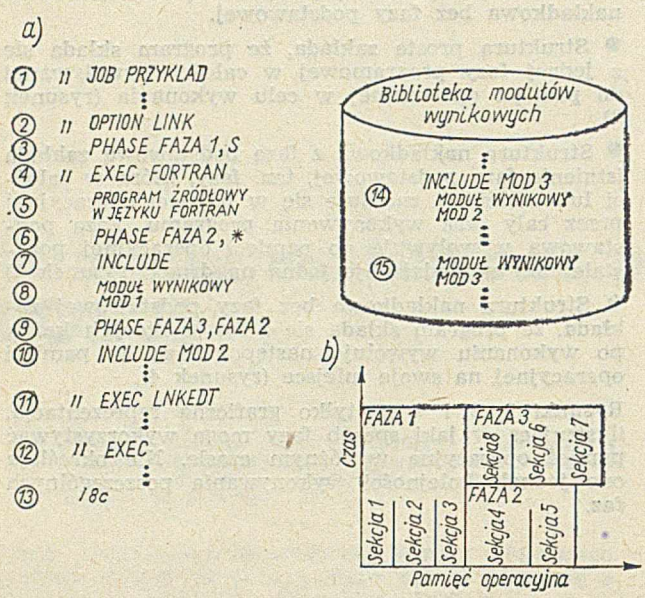


Rys. 3



Rys. 4

Samo formowanie zaprojektowanej struktury w systemie operacyjnym DOS/JS jest rzeczą stosunkowo prostą, opartą na odpowiednim wykorzystaniu kart sterujących dwóch typów: karty PHASE definiującej



Objaśnienia do rysunku 5 a):

1 — Karta indentyfikująca początek zadania o nazwie PRZYKŁAD w systemie DOS/JS.
2 — Karta zadająca reżim pracy systemu typu LINK.
3 — Karta identyfikująca początek tworzenia fazy FAZA 1 o początkowym adresie równym pierwszej wolnej pozycji w pamięci operacyjnej za obszarem systemowym.
4 — Karta powodująca wykonanie translatora języka FORTRAN.
5 — Program źródłowy w języku FORTRAN tworzący po translacji moduł wynikowy składający się z sekcji programowych SEKCJA 1, SEKCJA 2, SEKCJA 3.
6 — Karta identyfikująca początek tworzenia fazy FAZA 2 o początkowym adresie równym pierwszej wolnej pozycji w pamięci operacyjnej za fazą FAZA 1.
7 — Karta powodująca włączenie do fazy FAZA 2 modułu wynikowego z kart perforowanych następujących bezpośrednio po tej karcie.
8 — Karty modułu wynikowego składającego się z sekcji programowych SEKCJA 4, SEKCJA 5.
9 — Karta identyfikująca początek tworzenia fazy FAZA 3 o początkowym adresie w pamięci operacyjnej takim samym jak faza FAZA 2.
10 — Karta powodująca włączenie modułu wynikowego MOD2 z biblioteki modułów wynikowych do fazy FAZA 3.
11 — Karta powodująca wykonanie programu redagującego program PRZEKŁAD.
12 — Karta powodująca wykonanie pierwszej fazy zredagowanego programu PRZEKŁAD. Następne fazy tego programu są włączone do wykonania przez tą fazę — fazę FAZA 1.
13 — Karta identyfikująca koniec zadania dla systemu operacyjnego DOS/JS.
14 — Moduł wynikowy typu mieszanego zawierający zapis INCLUDE MOD 3 powodujący włączenie do modułu wynikowego MOD 2 modułu wynikowego MOD 3 oraz sam moduł wynikowy MOD 2 składający się z sekcji programowych SEKCJA 6, SEKCJA 7.
15 — Moduł wynikowy typu prostego MOD 3 składający się z jednej tylko sekcji programowej SEKCJA 8.

nazwę i początkowy adres danej fazy w pamięci operacyjnej oraz karty INCLUDE powodującej włączenie do budowanej fazy określonego modułu wynikowego z biblioteki modułów wynikowych, bądź z innego zewnętrznego nośnika informacji (np. z kart perforowanych) lub też niektórych tylko jego sekcji programowych. W skład tworzonej fazy mogą wejść także moduły wynikowe otrzymane bezpośrednio w procesie translacji programu źródłowego. Na rysunku 5 zilustrowano proces tworzenia określonej struktury nakładkowej programu w systemie operacyjnym DOS/JS.

Jak z powyższego wynika, przy projektowaniu struktury nakładkowej programu, należy przede wszystkim uwzględnić:

- ilość dostępnej pamięci operacyjnej
- Częstotliwość wykorzystywania każdej sekcji programowej
- Zależności pomiędzy sekcjami programowymi
- Sposób, w jaki sterowanie może być przekazywane pomiędzy fazami jednocześnie obecnymi w pamięci operacyjnej i fazami nakładającymi się.

Dopiero po rozpatrzeniu tych wszystkich czynników, programista jest w stanie zadecydować o ostatecznej strukturze swojego programu, której postać może być daleko bardziej skomplikowana aniżeli jakaś z trzech podstawowych struktur.

Na zakończenie warto podkreślić szczególne znaczenie tworzenia struktur nakładkowych programów, tyczących przetwarzania zbiorów danych. Wśród innych czynników, podstawowe znaczenie w systemie operacyjnym DOS/JS ze względu na szybkość przetwarzania zbiorów danych, maja tzw. współczynnik zblokowania i technika podwójnego buforowania operacji wejścia-wyjścia. Ze względu na szybkość korzystne jest stosowanie dużych współczynników zblokowania (jednocześnie rośnie efektywność wykorzystania zewnętrznych nośników informacji) oraz podwójnych buforów wejścia-wyjścia, co wiąże sie jednak z dużymi nakładami pamięci operacyjnej. I w takich właśnie przypadkach efektywnym rozwiązaniem, jest nadanie programom postaci odpowiedniej struktury nakładkowej.

Dialog Sokratyczny*)

O tym, jak niejaki Megma, najwyższy kapłan i Główny Projektant Systemu zdołał zrozumieć zalety Programowania Strukturalnego.

Megma: Sokratesie, chciałbym porozmawiać z tobą.
Sokrates: *Tak, Megmo? Rozmowa z tobą jest dla mnie zawsze przyjemnością. W czym mógłbym ci dopomóc?*

M. Otóż, Sokratesie, słyszałem, jak któregoś dnia w agorze mówiłeś dzieciom o Programowaniu Strukturalnym.

S: *Tak, to musiało być wczoraj. Ciekaw jestem co o tym myślisz?*

M: Nie chciałbym obrazić cię, Sokratesie, muszę jednak wyrazić moje uczucia.

S: *Zaiste, Megmo. Czyjeż uczucia możemy wyrażać, jeśli nie nasze.*

M: Muszę więc zakwestionować twoje nauki, Sokratesie. I mimo że zawsze myśli twoje są bardzo przekonywujące, nie mogą zgodzić się z twoimi naukami w tym jednym punkcie.

S: *Proszę więc, oświeć mnie, Megmo. Mylę się często i tylko dzięki podobnym do tej dysputom mogę poprawiać me błędy.*

M: Sokratesie, widzę zalety Programowania Strukturalnego. Daje ono czytelny kod, a być może nawet — jak twierdzisz — kod, który łatwo jest modyfikować. Muszę jednak stwierdzić, że nie sprawdziłem tego, jako że mój kod nigdy nie wymaga modyfikacji. Nie mogę się jednakże zgodzić z tobą, gdy twierdzisz, że idee te mogą być w codziennej praktyce programowania. Jasne jest bowiem, iż wszystkie techniki, które proponujesz są niezgodne z efektywnością. A nie twierdzisz przecież, że powinniśmy poświęcić efektywność dla piękna. Taki argument byłby dobry w sztuce, ale nie w dziedzinie bardziej doczesnej. Pitagorejczycy nauczają nas o pięknie trójkąta, ale nie sugerowaliby chyba, abyśmy robili koła wagonów z trójkątów równobocznych?

S: *Zbyt wiele pytań stawiasz naraz. Megmo. Czy mogę odpowiedzieć na nie po kolei, czy też muszę na wszystkie od razu?*

M: Jak wolisz, Sokratesie.

S: *Dziękuje ci. Myślę, że jeśli twój kod nigdy nie wymaga modyfikacji to zaiste musisz być błogosławiony przez bogów. Ale czy to nie ty napisałeś program, który odpowiada na pytania stawiane wyroczni w delfijskiej świątyni?*

M: To prawda. Uczyniłem to na żądanie samej wyroczni. Rozkaz przekazała mi kapłanka Eliza.

S: *Rzadki to honor. Ale czyż nie zdecydowano ostatnio, że wyrocznia daje zbyt pesymistyczne odpowiedzi?*

M: Tak. I niezbędnym okazało się przerobienie programu tak, by odpowiedzi były bardziej pogodne. Działo się to w zeszłym roku, Sokratesie, i wszyscy w Atenach wiedzą o tym. Nie bardzo wiem, do czego więc zmierzasz.

S: *Idźmy dalej. Jeżeli program ten musiał być przerobiony, czyż nie było to modyfikacją?*

M: Z pewnością.

S: *Ale powiedziałeś, że twoje programy nigdy nie muszą być modyfikowane. Muszę przyznać, że się w tym gubię.*

M: Wybacz, Sokratesie, Wy, filozofowie, jesteście zawsze tacy dokładni. Chodziło mi o to, że JA nigdy nie musiałem modyfikować moich programów.

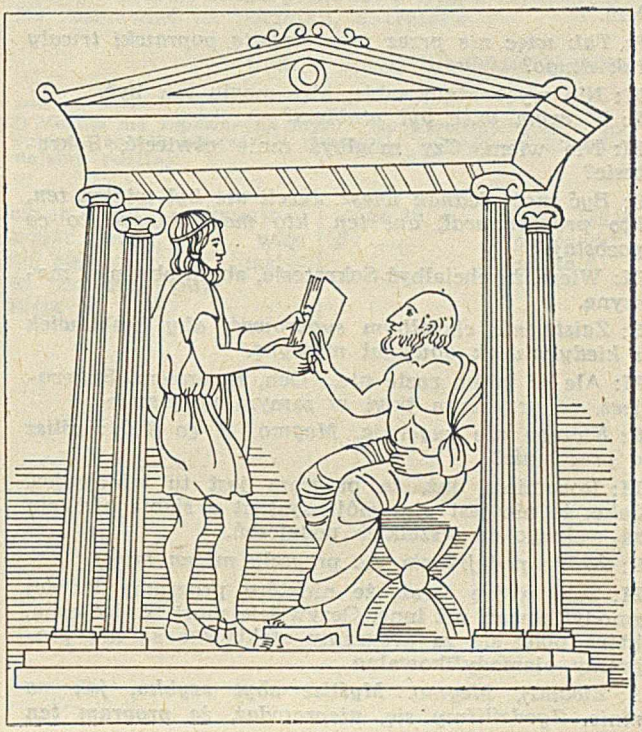
S: *Ale jak, w takim razie, były dokonywane poprawki? Czy sama wyrocznia podyktowała ci poprawny kod?*

M: Widzę, że żartujesz, Sokratesie. Wszyscy wiemy, że wyrocznia nie przeszła testu Turinga i nie może programować sama siebie. Po prostu zleciłem to memu asystentowi, Kobolowi. Znasz go na pewno, był w grupie, dla której wykładałeś.

S: *Tak, przyłączył się do grupy w zeszłym miesiącu. Nawiasem mówiąc, Megmo, widzę, że nie znasz moich metod. Nigdy nie wykładam, ponieważ jestem zbyt wielkim ignorantem. Tworzę te grupy tylko po to, aby zadawać pytania. W ten sposób mogę uczyć się od innych.*

M: Dziwna idea, Sokratesie. Jako Główny Projektant i najwyższy kapłan przywykłem do tego, że to do mnie ludzie przychodzą po wiedzę.

S: *Tak, rozumiem. Ale wracając do tematu, rozmawiałem z Kobolem i powiedział mi on, że twój kod był doskonalszy od wszystkiego, co widział dotychczas. Powiedział, że ten kod był szybki i prawie bezbłędny. Ale gdy dałeś mu tę pracę, był zaskoczony,*



*) DATAMATION, 1974, No 11.

ponieważ musiał napisać od nowa wiele części całego systemu, ale mogąc użyć twojego kodu.

M: Czy powiedział ci dlaczego, Sokratesie?

S: *Powiedział. Zdaje się, że twój kod był tak efektywny, że trudno było zauważyć, gdzie należy wstawić poprawkę. A ponieważ program napisany był w języku wyroczni, nie można było się zorientować, jak użyte są rejestry, ani też, jak przebiega sterowanie tego programu.*

M: Muszę przyznać, że nigdy nie skończyłem dokumentacji. Ale o co chodzi? Po cóż mamy asystentów, jeśli nie do takich prac?

S: *Myślę, że będę mógł odpowiedzieć na twoje pytanie, ale pozwól mi najpierw zadać inne. Co robiłeś, zanim stałeś się zamożny; z pewnością kiedyś nie miałeś asystentów? Jak radziłeś sobie wtedy?*

M: Otóż, Sokratesie, pierwszą pracę otrzymałem w Sparcie. Rząd wynajął mnie, abym napisał program, który wybiera ich króla. Gdy napisałem program, podjąłem zapłatę i przybyłem do Aten. Tak więc, raz tylko nie mogłem pozwolić sobie na asystentów. Jak wiesz, rząd Sparty płacił w tych czasach bardzo dobrze.

S: *Tak, ale ostatnio popadli w tarapaty. Słyszałem, że podobno mają obecnie dwóch króli.*

M: To prawda. Zdarzyło się to wkrótce po moim wyjeździe. Ciągle zastanawiam się, dlączego.

S: *Chciałem dowiedzieć się, czy zdarzyło się kiedyś, iż sam wykonałeś tę pracę, którą obecnie powierzasz swoim asystentom. Widzę jednak, że zbyt szybko stałeś się na to za bogaty. Spróbujmy więc inaczej. Czy twoje przedsiębiorstwo działa dobrze?*

M: Zaiste, działa.

S: *Czy nie zdarza się, że miewasz zbyt dużo pracy?*

M. Zdarza się, Sokratesie. Właśnie teraz jesteśmy opóźnieni z dużym projektem.

S: *Czy wiesz, jak to się stało? Myślę, że nie lubisz spóźniać się?*

M: O nie. To dziwny zbieg okoliczności. Kobol jest także z tym związany. Wykończenie projektu, o którym mówiliśmy wcześniej, zabrało mu tak dużo czasu, że nie mogłem go skierować do pracy nad nowym projektem.

S: *Twierdzisz więc, że gdyby Kobol mógł zmodyfikować program w krótszym czasie, nie byłbyś dzisiaj tak bardzo opóźniony.*

M: Wręcz przeciwnie. Jest chyba. najbardziej utalentowanym asystentem, jakiego kiedykolwiek miałem.

S: *Tak więc nie przez jego głupotę poprawki trwały tak długo?*

M: Nie wyobrażam sobie, aby mogło tak być.

S: *W czym więc był problem?*

M: Nie wiem. Czy mógłbyś mnie oświecić, Sokratesie?

S: *Być może jednak wiesz. Jeżeli nie był winien ten, kto programował, ani ten, kto modyfikował, to co pozostaje?*

M: Wiem, że chciałbyś Sokratesie, abym obwiniał maszynę.

S: *Zaiste nie chciałbym sugerować, aby ktokolwiek i kiedykolwiek obwiniał maszynę.*

M: Ale w takim razie nie... Och, rozumiem. Sugerujesz, że przyczyna tkwi w samym programie!

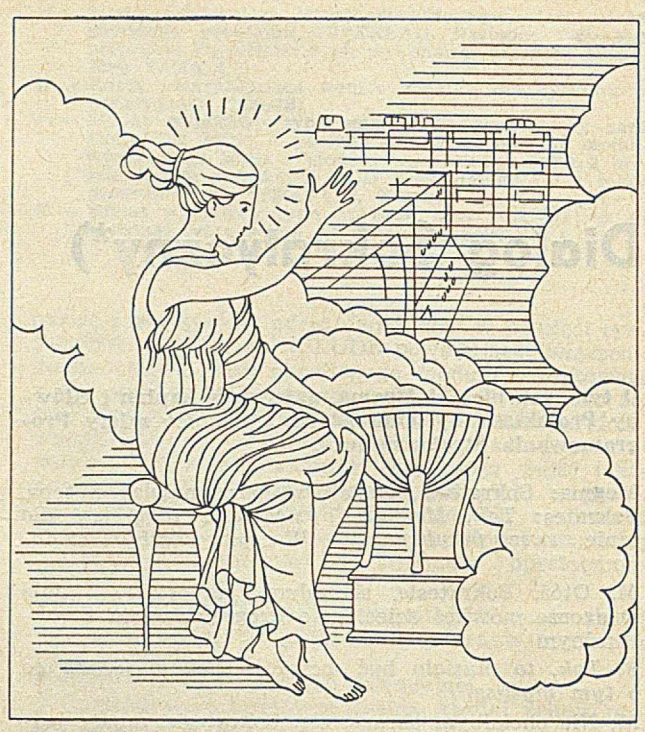
S: *Niczego nie sugeruję, Megmo. A co TY myślisz o programie?*

M: Rozumiem, już, że program jest tu winowajcą. Jako, że nie jest nim mój asystent a siebie uważam za wolnego od wszelkich podejrzeń.

S: *Żadne podejrzenia nie przyszły mi na myśl.*

M: Tak, widzę teraz, że niektóre programy łatwiej zmodyfikować, niż inne. Oczywiście musi istnieć cena, którą płaci się za programy efektywne są nią programy niemodyfikowalne.

S: *Zwolnij, Megmo. Myślisz zbyt szybko, jak na mnie. Zgodziliśmy się, nieprawdaż, że program ten musiał szczególnie ciężko poddawać się modyfikacjom. Ale czemu stwierdzasz od razu, że program nie może być jednocześnie efektywny i łatwy do modyfikacji?*



M: To z pewnością jasne dla ciebie, Sokratesie. Te dwie rzeczy nie mogą współistnieć w jednym programie. Pozwól mi rozumować tak, jak sam częstokroć czynisz — przez analogię. Weźmy więc na przykład ten nowy wyrób, który wprowadzano w zeszłym roku, ten ciemny produkt, który nazywają czekoladą. I weźmy inny produkt — mleko, które wszyscy dobrze znamy. Z pewnością widzisz, Sokratesie, że efektywność i modyfikowalność są jak te dwa produkty skrajnymi przeciwieństwami. Jeden jest czarny drugi biały, jeden stały a drugi ciekły. Niemożliwe jest ich łączenie w jakikolwiek sposób.

S: *Zgadzam się z tobą w sprawie produktów, Megmo, nie widzę jednak analogii, na której moglibyśmy oprzeć słuszny argument.*

M: Wytłumacz mi więc, Sokratesie, jak program może być jednocześnie modyfikowalny i efektywny.

S: *Nie mogę odpowiedzieć ci tak prosto, jak o to prosisz. Czuję, że nie wiem dość dużo na ten temat, zwłaszcza by przekonać takiego eksperta, jak ty. Wydaje ci się, że nie zgadzam się z tobą, ale to nie jest prawda. Czy mógłbyś mi może jaśniej wytłumaczyć, dlaczego te dwa terminy uważasz za niezgodne?*

M: Będę próbował, Sokratesie. Program efektywny musi być szybki. Musi oszczędnie używać swojej pamięci; nigdy nie może posługiwać się dwoma rejestrami tam, gdzie wystarczy jeden.

S: *Zgadzam się, że to gwarantuje efektywność programu.*

M: A po to, by program był modyfikowany, musi być — tak, jak nauczyłeś się strukturalny. A taki program musi być — zgodnie z twoim założeniem — czytelny. Musi być łatwo widoczne, co jest w danym punkcie wykonywane.

S: *Tak, zgadzam się z tobą. Ale ciągle nie widzę przeciwieństw. Gdy mówisz o efektywności, mówisz o sposobie, w jaki programu używa maszyna. Gdy masz na myśli modyfikacje, chodzi ci o programistę. W czym więc konflikt?*

M: Wiem już, skąd pochodzi twoja niepewność. Może przykład będzie tu pomocny. W programie wyroczni,

o którym mówiliśmy wcześniej, było wiele poprawek. Program został napisany w wewnętrznym języku wyroczni a potem wykryto błędy i trzeba było wstawić poprawki *). Wyobrażasz sobie, jak potem wyglądał kod, który i tak nie był przeznaczony dla prostych śmiertelników.

S: *Rozumiem, o co ci chodzi. Ale czyż nie miałeś zamiaru udokumentować ten program?*

M: Powiedziałem to.

S: *Czyż nie widzisz, że praca nad modyfikacją byłaby łatwiejsza, gdyby istniała dokumentacja programu?*

M: Nie widzę tego, Sokratesie. A to dlatego, że dokumentacja pokazywałaby wówczas idealny program z ustawionym w logiczny ciąg zbiorem czynności. A to nie pomogłoby nikomu w czytaniu prawdziwego programu z wstawionymi poprawkami.

S: *Megmo, pomyśl o tym, co właściwie powiedziałeś. Była w twojej wypowiedzi nispójność, której nie spodziewałem się po tak doświadczonym programiście, jak ty.*

M: Rozumiem już Sokratesie i twarz mą obleka rumieniec. Chodzi ci o to, że powinienem był przedstawić w dokumentacji taki program, jaki istnieje w rzeczywistości, a nie taki, jaki powinien istnieć. To dla mnie wielka rewelacja, Sokratesie. Myślę, że to ogromnie pomoże całemu przedsiębiorstwu. Nie wiem, jak ci mam dziękować.

S: *Poczekaj, Megmo. Myślę, iż w ogóle nie zasługuję na podziękowania, ponieważ zupełnie nie umiałem przekazać ci moich myśli.*

M: Nie jest tak, Sokratesie. Rozumiem wszystko o wiele lepiej.

S: *Może tak jest, ale twoje słowa zupełnie na to nie wskazują. Chyba, że znowu przeszkadza tu moja głupota.*

M: Sokratesie, jeśli cokolwiek z tego co powiedziałem nie odpowiada ci, powiedz, dlaczego. Czyż nie próbowaliśmy dojść do tego, jak program może być jednocześnie efektywny i modyfikowalny? i czyż nie objaśniłeś mi, że gdybym dobrze udokumentował mój kod osiągnąłbym oba te cele?

S: *Prawdą jest to, co mówisz, ale przypominasz mi jednocześnie Persów, którzy tak spiesznie przechodzili koło Maratonu, że nie zauważyli naszych żołnierzy z drugiej strony.*

M: Pokaż mi więc, gdzie błądzę, Sokratesie.

S: *Nie błądzisz, Megmo. Po prostu nie przykładasz wagi do swoich słów. Powiedziałeś, że dokumentacja, którą masz teraz pokazuje idealny, logicznie powiązany program. Powiedziałeś też, że prawdziwy program zawiera wiele poprawek, które różnią go od ideału.*

M: Tak powiedziałem.

S: *I tu przestaję rozumieć. Dlaczego kod nie wygląda tak, jak dokumentacja?*

M: Jestem zgubiony, Sokratesie. Nie mogę zrozumieć, o czym mówisz.

S: *To proste Megmo. Czy kod nie może mieć logicznie uporządkowanej postaci? Czy podobne czynności nie mogą zostać zgrupowane? Innymi słowy, czy tworząc dokumentację logicznie i stopniowo, nie możesz pisać kodu w ten sam sposób?*

M: Wiemy już do czego zmierzasz, Sokratesie. To zrozumiały błąd u osoby świeckiej, którą jesteś. Nie będąc programistą, nie wiesz, jak jest tworzony kod. Otóż po pierwsze, piszę zarys końcowego programu. Jest to, jak powiedziałbyś, pierwsze przybliżenie. Następnie testuję ten kod. Gdy znajduję błąd, wstawiam poprawkę. W ten sposób, gdy program jest gotowy, nie tworzy już — tak jakbyś pragnął — logicznego ciągu.

S: *Ale czy...*

M: Wiem, co chcesz powiedzieć. Dlaczegóż by nie przepisać kodu tak, aby naśladował dokumentację? Musisz wiedzieć, Sokratesie, że nasza religia programistów zabrania nam dotykać pracującego kodu.

S: *Nie o to chciałem zapytać, Megmo. Chodziło raczej o rzecz następującą: czy nie możesz, napisać od razu kompletnego kodu? Pozwól zapytać, czy wstawiasz też poprawki w miejsce błędów syntaktycznych?*

M: Nie, Sokratesie. W ten sposób poprawiam tylko błędy logiczne.

S: *A czy nie można by napisać kodu w ten sposób, aby w pierszym „przyliżeniu" były tylko błędy syntaktyczne? Czy w ten sposób nie można by połączyć modyfikowalności z efektywnością?*

M: Widzę wielką światłość, Sokratesie. Sugerujesz, że po to, aby osiągnąć nasz cel, powinniśmy staranniej i dokładniej projektować. Widzę, że to jest dobrym rozwiązaniem problemu. Bo gdybyśmy poświęcili więcej czasu projektowaniu, otrzymalibyśmy — używając mojej analogii — coś w rodzaju czekoladowego mleka!

S: *I czyż czas, stracony przy projektowaniu, nie został nadrobiony przy uruchamianiu?!*

M: Zaiste, Sokratesie. Rozmowa nasza jest dla mnie wielkim oświeceniem. Chciałbym teraz oddalić się i przemyśleć to, o czym mówiliśmy.

S: *Byłem szczęśliwy, mogąc rozmawiać z tobą, Megmo.*

M: Obawiam się, czy aby rozmowa ze mną nie zawiodła cię, Sokratesie?

S: *Dlaczego tak myślisz?*

M: Mieliśmy przecież rozmawiać o programowaniu strukturalnym, a doszliśmy tylko do tego, jak powinienem pisać lepsze programy. Być może kiedy indziej będziemy mogli pomówić o tym, o czym chciałeś rozmawiać na początku, Sokratesie.

tłum. i oprac. **Stanisław Matwin**

---

*) Megma ma zapewne na myśli rozpowszechniony w starożytności sposób wstawiania poprawek, którego przykład podajemy poniżej:

chcemy dokonać pewnej poprawki pod etykietą α	postępujemy więc jak następuje	α : **go to** ω β :	ω: „tekst poprawki" **go to** β

Formy organizacyjne pionów informatyki*)

Zadaniem służb projektowo-programowych jest realizowanie pięciu następujących funkcji:

- projektowanie nowych systemów informatycznych
- spełnianie nadzoru autorskiego nad systemami eksploatowanymi (konserwacja systemów)
- szkolenie i doskonalenie personelu fachowego
- tworzenie i wdrażanie standardów pracy
- prowadzenie prac badawczych i studialnych w zakresie nowoczesnych metod pracy i nowoczesnego sprzętu

Organizacja pionu projektowo-programowego winna sprzyjać realizowaniu tych funkcji. Można wyróżnić cztery zasadnicze typy struktur organizacyjnych występujących w pionie projektowo-programowym: zorientowana problemowo, funkcjonalna, projektowo-funkcjonalna i projektowo-sztabowa

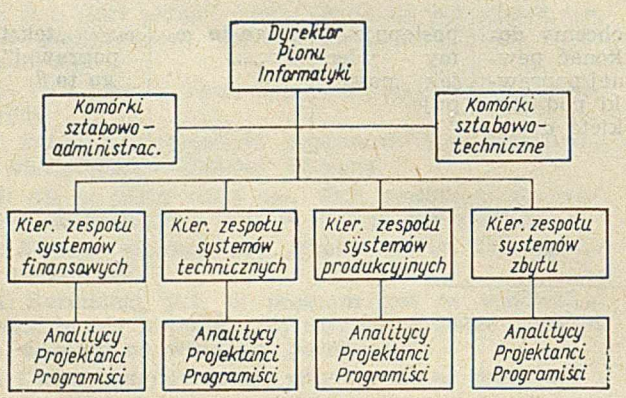
Organizacja zorientowana problemowo

W tym typie organizacji dla każdej z dziedzin zastosowań tworzy się zespół, składający się z analityków, projektantów i programistów (rys. 1). Kierownik zespołu odpowiada za projektowanie i wdrażanie nowych systemów oraz za sprawowanie nadzoru autorskiego nad systemami swej dziedziny. On też rozdziela pracę pomiędzy podległych mu pracowników.

Doskonaleniem i szkoleniem pracowników, wprowadzaniem nowych standardów pracy, pracami badawczymi nad nowymi metodami pracy i nowym sprzętem — zajmują się komórki sztabowe pionu.

Zaletą takiej organizacji jest jej orientacja na użytkownika. Pracownicy poszczególnych zespołów projektowo-programowych stają się wkrótce specjalistami od problemów jakie występują u użytkownika. Ten typ organizacji w zakładowym ośrodku obliczeniowym jest korzystny w początkach wdrażania informatyki.

W miarę zwiększania się liczby systemów odcinkowych organizacja ta jednak staje się przestarzała; pojawiająca się potrzeba integracji systemów sprawia, że niektóre z grup zastosowań tracą rację bytu i konieczne staje się elastyczne modyfikowanie tej organizacji np. poprzez utworzenie nowej grupy zastosowań — zespołu dla integracji systemów.

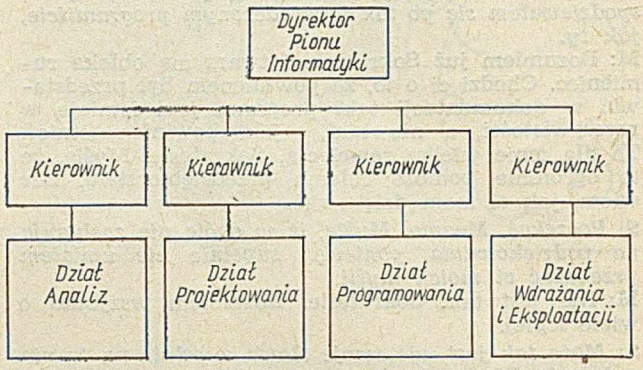


Rys. 1. Schemat struktury organizacji zorientowanej problemowo

*) Na podstawie artykułu T. R. Gildersleeve „Organizing the Data Processing Function" zamieszczonego w DATAMATION, vol. 20, nr 11, 1974.

Organizacja funkcjonalna

Celem organizacji funkcjonalnej jest zwiększenie wydajności i jakości pracy wykwalifikowanej kadry specjalistów poprzez zawężenie ich specjalizacji. Problemy doskonalenia zawodowego, wprowadzanie standardów pracy, prace badawcze — stają się częścią



Rys. 2. Schemat organizacji funkcjonalnej

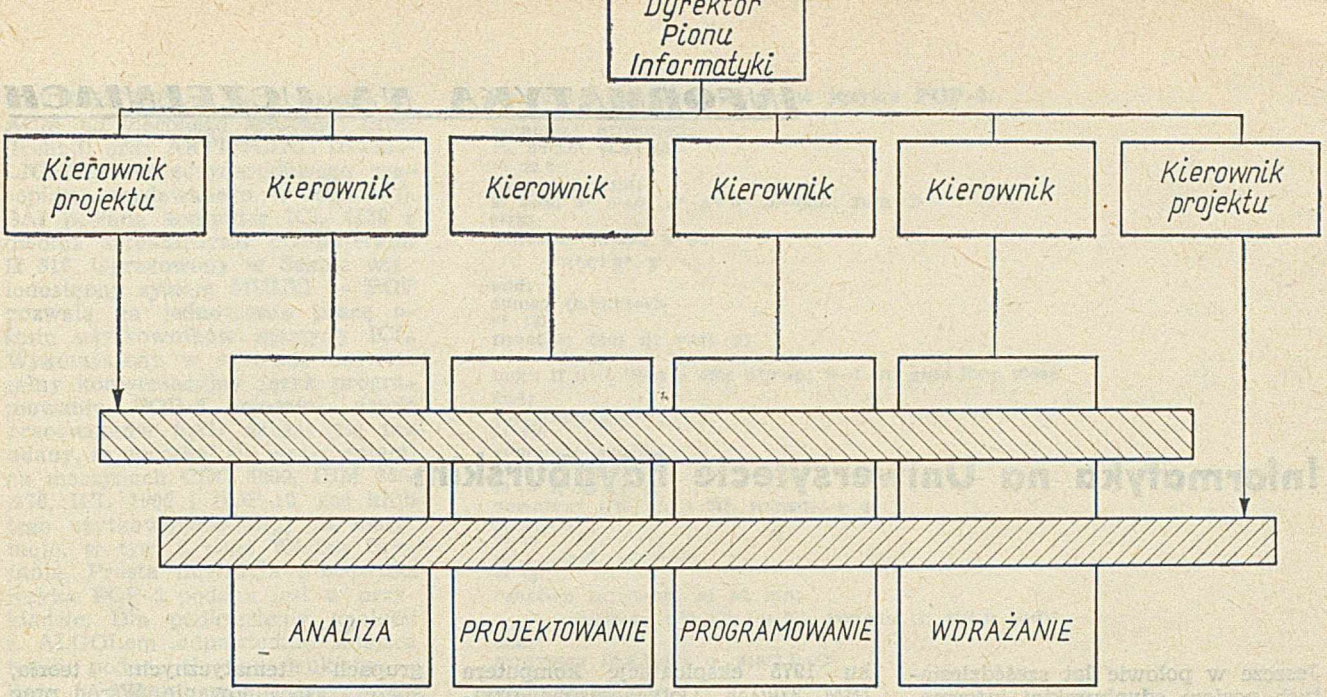
zadań każdego z kierunków komórek funkcjonalnych. Jest wiele możliwych wariantów tej organizacji. Może to być wariant przedstawiony na rys. 2, może również nastąpić połączenie analizy z projektowaniem lub analizy z wdrażaniem. W tym typie organizacji odpowiedzialność za nowy system informatyczny zostaje rozdrobniona pomiędzy poszczególne działy funkcjonalne. Na przykład dział analiz ponosi odpowiedzialność za wstępną koncepcję systemu. Opracowanie analityków (w postaci założeń funkcjonalnych systemu) jest punktem wyjścia dla prac działu projektowania (odpowiedzialnego za techniczne rozwiązania systemowe ujęte w projekcie technicznym systemu). Dalszą część odpowiedzialności za ten sam system podejmuje dział programowania (poprawność działania programów i ich zgodność z założeniami do programów wynikających z projektu technicznego i założeń funkcjonalnych). Pozostała wreszcie część odpowiedzialności za system rozkłada się na dział przetwarzania (wdrożenie i eksploatacja systemu). Takie rozdrobnienie odpowiedzialności za system utrudnia kontrolę przebiegu prac nad systemem, zwłaszcza w przypadkach uchybień w jakości, lub gdy zachodzi potrzeba — modyfikację systemu.

Organizacja projektowo-funkcjonalna

W organizacji projektowo-funkcjonalnej (rys. 3) realizowane są, przy zachowaniu struktury fukcjonalnej naszej organizacji, określone zadania projektowe.

Każdy z kierowników funkcjonalnej komórki zachowuje swoje uprawnienia w zakresie szkolenia i doskonalenia personelu, opracowywania i wdrażania standardów oraz prowadzenia prac badawczo-rozwojowych.

Natomiast dla każdego podejmowanego projektu (czy będzie to zaprojektowanie nowego systemu, czy konserwacja systemu już eksploatowanego) wyznacza się kierownika projektu *(project leader)*, którego zadaniem jest nadzorowanie całości realizacji projektu. Najistotniejszym w tym typie organizacji jest to, że kierownicy projektu nie podlegają kierownikom funkcjonalnym, lecz bezpośrednio kierownictwu pionu.



Rys. 3. Schemat organizacji projektowo-funkcjonalnej

Powodem, dla którego występują oddzielne linie zależności jest to, iż niekiedy cele kierowników funkcjonalnych i projektów stają się konfliktowe. Tylko rozdzielenie linii zależności gwarantuje zwierzchniemu kierownictwu uzyskanie dostatecznie obiektywnych informacji przy rozstrzyganiu sytuacji konfliktowych w sposób najwłaściwszy z punktu widzenia interesów całego pionu informatycznego.

Organizacja zespołów projektowych może przyjąć jedno z następujących trzech rozwiązań:

1. W skład zespołu projektowego na czas opracowywania danego projektu wchodzą pracownicy z różnych działów funkcjonalnych: analitycy, projektanci i programiści, podlegający w tym czasie kierownikowi projektu, który planuje i kontroluje ich pracę.
2. Planowanie i kierowanie pracą podległych pracowników znajduje się w gestii kierowników funkcjonalnych.

Zadaniem kierownika projektu jest dopilnowanie, aby planowanie prac nad danym projektem, a następnie realizacja tych prac w poszczególnych działach funkcjonalnych były wzajemnie skoordynowane w sposób umożliwiający ukończenie zadania z ustalonym wykazem tematów w planowanym terminie i przy nieprzekroczeniu planowanych nakładów.

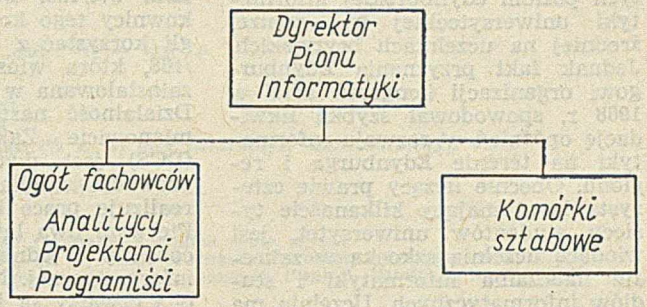
3. Kombinacja obu powyższych rozwiązań: część zespołu projektowego podlega bezpośrednio kierownikowi projektu, a część mimo iż również pracuje nad tym samym tematem, podlega swemu kierownikowi funkcjonalnemu.

Typ pierwszy jest najwłaściwszy dla małych projektów, angażujących jednorazowo nie więcej niż 15—20 osób. Drugi jest typowy dla dużych projektów — przy których zatrudnia się setki ludzi. Trzeci jest najodpowiedniejszy dla projektów średniej wielkości. Kierownik projektu swą kierowniczą funkcję spełnia jedynie w okresie prac nad projektem. Z chwilą zatwierdzania projektu (zadania) — kończy się również funkcja kierownika projektu, który wraca na swe poprzednie miejsce pracy jako analityk, projektant czy programista.

Celem takiej organizacji pracy jest wyciągnięcie korzyści z zalet organizacji funkcjonalnej, z jednoczesnym uniknięciem jej wad. Element funkcjonalny w tej organizacji korzystnie wpływa na podnoszenie kwalifikacji przeciętnych specjalistów, zaś element projektowy (zadaniowy) stanowi narzędzie sprawnego sterowania postępem prac.

Organizacja projektowo-sztabowa

Na komórkach sztabowych spoczywają funkcje związane z doskonaleniem zawodowym pracowników, podnoszeniem ich kwalifikacji, opracowywaniem stan-



Rys. 4. Schemat organizacji projektowo-sztabowej

dardów z prowadzeniem prac badawczo-rozwojowych (rys. 4).

Dla każdego projektu tworzony jest spośród ogółu fachowców, odpowiedni zespół składający się z kierownika tematu, analityków, projektantów i programistów.

Ten typ organizacji może mieć zastosowanie wszędzie tam, gdzie zespół fachowców jest dojrzały, lecz nie dość liczny, aby utworzyć organizację projektowo-funkcjonalną. Słabością tej formy organizacyjnej jest fakt, że komórki sztabowe nie mają bezpośredniego wpływu (w zakresie wdrażania standardów) na tych, którzy mają się do nich stosować.

Nadzór nad stosowaniem standardów spoczywa w dużej mierze na kierownictwie naczelnym. Odnosi się to również do organizacji zorientowanej problemowo (wdrażanie standardów należy w nich do zadań komórek sztabowych).

Wadą organizacji projektowo-sztabowej jest także to, że zadania przydziela poszczególnym pracownikom naczelne kierownictwo pionu informatycznego.

* * *

Jest oczywistym, że dobry pracownik zawsze będzie wykonywał prawidłowo i efektywnie swoje zadania bez względu na strukturę organizacyjną, w jakiej się w danej sytuacji znajduje. Prawidłowa struktura organizacyjna, tzn. adekwatna do celów i funkcji jakie realizować ma dana jednostka organizacyjna, stanowi jednak istotne, a często niedocenione źródło wzrostu wydajności pracy, umożliwiające przemianę dobrego pracownika w super-wydajnego, przeciętnego w wyjątkowego i miernego w wydajnego. Toteż wydaje się że warto ocenić własną organizację i ewentualnie dokonać wyboru najwłaściwszej, w danych warunkach struktury organizacyjnej, dostosowanej do realiów konkretnego przedsiębiorstwa lub instytucji.

Tłum i oprac. **Ludwik J. Rossowski**

Informatyka na Uniwersytecie Edynburskim

Jeszcze w połowie lat sześćdziesiątych poziom edynburskiej informatyki uniwersyteckiej był poniżej średniej na uczelniach brytyjskich. Jednak fakt przyznania Edynburgowi organizacji kongresu IFIP w 1968 r. spowodował szybką likwidację opóźnień w rozwoju informatyki na terenie Edynburga i regionu. Obecnie liczący prawie czterysta lat i mający kilkanaście tysięcy studentów uniwersytet, jest wiodącą uczelnią szkocką w zakresie nauczania informatyki i studiów informatycznych. Uczelnia ma trzy ośrodki informatyki: Centrum Obliczeniowe *(Edinburgh Regional Computing Centre* — ERCC), Zakład Informatyki *(Department of Computer Science* — DCS) i Szkołę Sztucznej Inteligencji *(School of Artificial Intelligence* — SAI).

Czytelnicy mieli już okazję zapoznać się z pierwszym ośrodkiem (INFORMATYKA 10/1974). ERCC otrzyma we wrześniu 1975 roku komputer ICL 2980, największy model z nowej serii maszyn firmy ICL. Jako podstawowe oprogramowanie ma być gotowe do końca 1975 roku. Konfiguracja przyszłego systemu jest następująca: jednostka centralna: ICL 2980, pamięć operacyjna 4×512 Kbajtów, pamięć dyskowa 3 jednostki po 6 Mbajtów na dyskach niewymiennych, 16 jednostek po 100 Mbajtów na dyskach wymiennych, pamięć taśmowa: 2 jednostki z zapisem 9 ścieżkowym i prędkością zapisu/odczytu 320 Kb/s, 4 jednostki z zapisem 7 ścieżkowy i prędkością zapisu/odczytu 120 Kb/s, 2 jednostki z zapisem 9 ścieżkowym i prędkością zapisu odczytu 120 Kb/s, drukarka wierszowa 1, czytnik kart — 1, perforator kart — 1, konsola operatora — 2, procesor telekomunikacyjny ICL 7905.

Zdaniem producenta — maszyny serii ICL 2900 utrzymają się w eksploatacji do końca naszego stulecia. Przyszli użytkownicy edynburscy szacują, że nowy system będzie miał moc obliczeniową co najmniej trzykrotnie taka, jaką ma IBM 370/ /158. W związku z otrzymaniem ICL 2980 Centrum zakończy w roku 1975 eksploatację komputera IBM 370/158. Dotychczasowi użytkownicy tego komputera będą mogli korzystać z maszyny IBM 370/ /168, która wiosną 1975 r. będzie zainstalowana w Newcastle.

Działalność następnego ośrodka, a mianowicie Zakładu Informatyki (DCS) jest dwukierunkowa. DCS prowadzi studia informatyczne i realizuje prace badawcze.

Pierwsze dwa lata studiów poświęcone są w jednej trzeciej tematyce informatycznej. Natomiast rok trzeci i czwarty są już w całości przeznaczone na informatykę (jeżeli jest ona wyłączną specjalizacją studenta), lub w równych częściach na informatykę i drugą specjalność. Studenci mogą bowiem kończyć jednocześnie przykładowo informatykę i matematykę, albo informatykę i fizykę. Pierwsze dwa lata studiów informatycznych są dostępne również dla studentów innych specjalności. W programie studiów nacisk kładzie się na projektowanie i praktyczną realizację systemów komputerowych.

Przez cały okres studiów student ma dostęp do komputera ICL 4-75 w systemie wielodostępnym EMAS *(Edinburgh Multi Acces System* — Edynburski System Wielodostępny), umożliwiającego realizację prac z programowania. Poczwąwszy od drugiego roku studiów korzysta on również z małych komputerów Zakładu Informatyki.

Są to: 2 maszyny PDP-7, PDP-8 i PDP-15 (produkcji firmy DIGITAL EQUIPMENT CORPORATION), NOVA (produkcji firmy DATA GENERAL) oraz INTERDATA 74 (hierarchiczny system 2 procesorów programowych z kilkunastoma procesorami podporządkowanymi). W ten sposób studenci poznają sprzęt firm wiodących w dziedzinie produkcji minikomputerów. Na trzecim i czwartym roku studiów studenci mają również zajęcia laboratoryjne z zakresu obsługi technicznej sprzętu liczącego. Równolegle do Informatyki stosowanej wykładana jest informatyka teoretyczna. Od drugiego roku zajęcia przebiegają w trzech równych czasowo grupach tematycznych: teoria, sprzęt, programowanie. Wśród prac badawczych realizowanych obecnie w Zakładzie na czoło wysuwa się projektowanie wspomagane komputerem *(Computer Aided Design* — CAD). Jest to główny problem węzłowy, a praca nad jego realizacją przewidziana jest na kilka lat. Do dyspozycji zespołów pracujących nad problemem CAD oddano komputer PDP-10, mający dwa procesory (system rozbudowano latem 1974 roku) i wyposażony między innymi w ekranopisy, zarówno znakowe, jak i graficzne, oraz pisaki. Satelitami PDP-10 są: minikomputer NOVA i PDP-7. Oprócz tego system PDP-10 współpracuje z maszyną PDP-11, będącą własnością innej uczelni edynburskiej, z którym ma bezpośredni dostęp poprzez dwie linie telefoniczne. W ten sposób jest lepiej wykorzystana duża moc obliczeniowa systemu działającego w sposób ciągły.

Szkoła Sztucznej Inteligencji (SAI) prowadzi prace naukowo-badawcze głównie o charakterze teoretycznym.

W skład SAI wchodzą cztery zakłady: Zakład Logiki Rachunkowej *(Department of Computational Logic),* Zakład Maszynowej Inteligencji *(Department of Machine Inteligence),* Zakład Psychologii *(Theoretical Psychology Unit)* i Laboratorium Bioniki *(Bionics Research Laboratory).* Rozpracowywana w Szkole tematyka jest bardzo różnorodna: automatyczne dowodzenie twierdzeń, rachunek predykatów jako język programowania, dowodzenie poprawności programów, semantyka języków programowania, gry komputerowe, programowany robot, analiza i synteza obrazów, automatyczna analiza elektroencefalogramów. Publikacje naukowe pracowników i doktorantów SAI wydawane są w kolejnych tomach wydawnictwa Machine Inteligence. Pracownicy Szkoły są także redaktorami dwóch czasopism: ARTIFICIAL INTELIGENCE and SIMULATION of BEHAVIOUR — EUROPEAN NEWSLETTER (odpowiednik amerykańskiego SIGART

NEWSLETTER — dwumiesięcznika ACM poświęconego sztucznej inteligencji) oraz ARTIFICIAL INTELLIGENCE (międzynarodowego czasopisma wydawanego w Holandii). SAI posiada komputer ICL 4130 z dwoma satelitarnymi komputerami H 316. Opracowany w Szkole wielodostępny system MULTI — POP pozwala na jednoczesną pracę ośmiu użytkowników maszyny ICL. Wykorzystany w systemie uniwersalny konwersacyjny język programowania POP-2 (również dzieło pracowników SAI) okazał się tak udany, że doczekał się inplementacji na maszynach CDC 6000, IBM 360/ /370, ICL 1900 i PDP-10, zaś klub jego użytkowników liczy 24 instytucje, w tym 8 poza Wielką Brytanią. Prosta ilustracja możliwości języka POP-2 podana jest w przykładzie. Dla podkreślenia analogii z ALGOLem odpowiednie miejsca tekstu podano tłustym drukiem.
Poza wierszami zaczynającymi się od **, a będącymi wydrukami wartości ze stosu, cały pozostały tekst był pisany przez programującego w trybie konwersacyjnym. → oznacza prześlij wartość ze stosu, zaś =⇒ drukuj wartości ze stosu kasując stos.
Na zakończenie ciekawostki o informatyce uniwersyteckiej. Architekci dysponują własnym, odpowiednio wyposażonym i oprogramowanym komputerem. Ekonomiści uczą się zarządzania przedsiębiorstwami uczestnicząc w komputerowych grach gospodarczych. Lingwiści posiadają programowany syntezator mowy ludzkiej. Komputer PDP-8 wyposażony jest w program syntezy czterogłosowej muzyki. Z komputerem PDP-10 można grać w szachy. Użytkownicy PDP-15 operują programowo na obrazach dających wrażenie trójwymiarowości.

Przykład. Programowanie w języku POP-2.

```
comment arithmetic;
12. 0+2.5* (1.5+2.5)
** 22.0
vars a b sum;
2* 2→a; 3* a→b; a* a+b* b→sum; sum=>
**160
function sumsq x y;
      x*x+y* y
end;
sumsq (a,b)+1=>
** 161
function fact n; vars p;
      1→p;
loop: if n=0 then p else n*p→p; n—1→n; goto loop close
end;
fact (fact/3)=>
** 720
comment arrays;
vars a i j;
10 → i; 20 → j;
newarray ([% 1,i,1,j %], sumsq) → a;
a(2,3)=>
** 13
10 → a(2,3); a(2,3)=>
** 10
function arraysum a1 a2 mn;
      newarray ([% 1,mn,1,n %], lambda ij; a1(i,j) end)
end;
arraysum (a,a,10,20) → a;a(2,3)=>
** 20
comment lists;
vars u;
1 → i; 2 → j;
[% i, i+j, „dog", „cat" %] → u;u=>
** (1 3 dog cat)
cons („pig", u)=>
** [pig 1 3 dog cat]
function append x y;
      if null(y) then (% x %) else cons(hd)y), appednd(x,tl(y)) close
end;
append (4, [% 1, i+1, 3%]=>
** [1 2 3 4]
comment records;
vars consper destper forename surname male p1 p2;
recordfns ("person", [0 01]→male→surname→forename→destper→consper;
consper ("jane", "jones", false)→p1; consper ("sam", "smith", true) → p2;
surname (p1)=>
** jones
datalist(p1)=>
** [jane jones 0]
function marry x y
      if male (x) and not (male(y)) then surname(x) → surname(y) close
end;
marry (p2, p1); datalist (p1)=>
** [jane smith 0]
```

Gerard Zieliński
Instytut Matematyki
Politechniki Warszawskiej

Zmiany organizacyjne w informatyce bułgarskiej

Z dniem 1 maja 1974 r. powołane zostało w Ludowej Republice Bułgarii Zjednoczenie Naukowo-Produkcyjne AWTOMATYZACJA, podległe bezpośrednio Radzie Ministrów. W skład Zjednoczenia AWTOMATYZACJA weszły następujące jednostki organizacyjne:

- Zjednoczenie MOSI obejmujące sieć rejonowych ogólnodostępnych ośrodków obliczeniowych — podległe uprzednio Ministerstwu Informacji i Łączności.

- Centralny Instytut Naukowo-Badawczy i Rozwojowy Metod Projektowania i Wdrażania Zautomatyzowanych Systemów Zarządzania i Automatyzacji Przygotowania Produkcji ORGPROJEKT wraz z filią w Ruse — podległe uprzednio Ministerstwu Przemysłu Maszynowego.

- Naukowo-Badawczy i Projektowy Instytut Projektowania Automatyki NIPIA — podległy uprzednio Ministerstwu Przemysłu Maszynowego.

Instytuty ORGPROJEKT i NIPIA połączone zostały w jeden Centralny Instytut Naukowo-Badawczy i Projektowy Kompleksowej Automatyzacji CNIKA. W Instytucie CNIKA znajduje się centralna biblioteka programów komputerowych.
Zjednoczenie AWTOMATIZACJA ma cztery piony organizacyjne:
— badań i rozwoju
— postępu technicznego
— planowania, ekonomiki, finansów i kadr
— organizacyjny.
Poszczególnymi pionami kierują Z-cy Dyrektora Naczelnego.

Dyrektorowi Naczelnemu Zjednoczenia AWTOMATIZACJA podlega bezpośrednio 28 usługowych ośrodków komputerowych, Instytut CNIKA oraz Ośrodek Szkolenia i Kadr.
W pionie Z-cy Dyrektora Naczelnego d/s badań i rozwoju znajdują się m.in. zagadnienia prognozowania i planowania rozwoju informatyki. Pion Z-cy Dyrektora Naczelnego d/s postępu technicznego zajmuje się zagadnieniami eksploatacji komputerów, konserwacji i remontów oraz inwestycji.
Zagadnienia dotyczące informatycznych systemów zakładowych i branżowych pozostały w branżowych ośrodkach naukowych i w branżowych ośrodkach ETO.
W ośrodkach usługowych Zjednoczenia AWTOMATYZACJA pracuje około 50 komputerów w tym 7 typu MIŃSK-22 i 3 MIŃSK-32. Ponadto w ośrodkach tych pracuja 2 komputery typu ODRA-1204, 1 IBM 360/30, 1 R-20 i 1 R-30. (tw)

Problemy wielodostępnych systemów informatycznych

Krajowa Konferencja Naukowa, Wrocław, 6—7.II.1975 r.

Konferencja naukowa poświęcona problemom wielodostępnych systemów informatycznych została zorganizowana w Polsce po raz pierwszy. Rozwój sprzętu komputerowego oraz konieczność jego efektywnego użytkowania w naszym kraju dopiero teraz zmusiła informatyków do kompleksowego zainteresowania się systemami wielodostępnymi.

Konferencja zorganizowana przez Politechnikę Wrocławską, Polski Komitet Automatycznego Przetwarzania Informacji (oddział wojewódzki we Wrocławiu) oraz Towarzystwo Naukowe Organizacji i Kierowania (oddział we Wrocławiu) pozwoliła na zapoznanie się uczestników (około 600 osób) *) z problemami oraz obecnym stanem systemów wielodostępnych w Polsce.

Goście z Czechosłowacji, NRD, Węgier, Hiszpanii, Wielkiej Brytanii oraz USA zaprezentowali funkcjonowanie i kierunki rozwoju systemów wielodostępnych oraz sieci komputerowych w ośrodkach krajowych, przeważnie związanych z uczelniami wyższymi.

W konferencji zapowiedzieli swój udział przedstawiciele ośrodków fraucuskich i austriackich, nie przybli oni jednak na obrady (ich referaty znajdują się w materiałach konferencji).

Program konferencji przewidywał cztery referaty problemowe otwierające pierwszy i drugi dzień obrad:

— Aktualne problemy systemów wielodostępnych
— Matematyczne podstawy systemów wyszukiwania informacji
— Matematyczne modele organizacji banku danych (nie wygłoszony)
— Problemy obsługi zgłoszeń w systemach wielodostępnych.

Obrady o charakterze plenarnym podzielono na cztery sesje tematyczne:

- ZAGADNIENIA OGÓLNE
- BADANIE I PROJEKTOWANIE SYSTEMÓW WIELODOSTĘPNYCH
- TRANSMISJA DANYCH W SYSTEMACH WIELODOSTĘPNYCH
- ZASTOSOWANIA.

Początkowo zamiarem organizatorów było prowadzenie obrad w sesjach tematycznych równolegle, ale duże zainteresowanie uczestników całością problematyki spowodowało przyjęcie plenarnego charakteru obrad. Taki przebieg konferencji nie pozwolił na rozwinięcie się dyskusji, pozostawiając jedynie czas na krótkie pytania pod adresem autorów.

Wszystkie referaty zostały zamieszczone w materiałach konferencyjnych i wcześniej rozesłane uczestnikom, co umożliwiło tym, do których materiały dotarły, zapoznanie się z ich treścią.

Większość referatów dotyczących pracujących systemów wielodostępnych przedstawiała nabyte doświadczenia bądź problemy powstałe przy eksploatacji systemów: SA CYFRONET (na komputerze CYBER-72 w Instytucie Badań Jądrowych Otwock-Świerk), POLRAX-2 (na komputer ODRA 1305 w ZETO Wrocław) oraz WASC (na ODRĘ 1305 w Politechnice Wrocławskiej).

Jeden z referatów dotyczył doświadczeń wynikających z eksploatacji systemu zdalnego przetwarzania w Ministerstwie Przemysłu Maszynowego (IBM 370).

Szkoda, że zabrakło przedstawiciela Przemysłu Okrętowego, który podzieliłby się doświadczeniami z eksploatacji wielodostępnego systemu pracującego na ICL SYSTEM 4/70 w ZIPO w Gdańsku-Oliwie.

W tematyce dotyczącej zagadnień ogólnych, ciekawy referat pt. „Zagadnienia przydziału pamięci operacyjnej w wielodostępnych systemach cyfrowych" wygłosił Tadeusz Mika z Zakładu Informatyki Politechniki Wrocławskiej przedstawiając interesujący, choć nie sprawdzony w praktyce, model przydziału.

W referatach zarówno polskich, jak i zagranicznych podkreślano konieczność prowadzenia badań z dziedziny wielodostępu, zwracano uwagę na fakt, że koszty takich badań zwracaja się bardzo szybko. Charakterystyczne dla tej dopiero rozwijającej się w Polsce dziedziny było wezwanie prof. Z. Pawlaka z Centrum Obliczeniowego PAN do stosowania naukowych metod w rozwiązywaniu jej problemów, by np. nie poszukiwać nieistniejacych rozwiązań, których nieistnienia można dowieść na gruncie matematyki. Przedstawiciel Wielkiej Brytanii w swoim referacie sygnalizował znamienny fakt, iż nawet komputer dużej mocy nie sprosta zadaniom występującym np. na niewielkiej (ok. 2000 osób) uczelni wyższej. Dopiero sieć komputerowa jest w stanie zrealizować wszystkie postawione zadania.

W godzinach popołudniowych pierwszego dnia obrad uczestnicy mieli możność zapoznania się w Zakładach MERA-ELWRO z komputerem ODRA 1305 w maksymalnej konfiguracji lokalnej z pamięcią 128 K oraz pamięciami dyskowymi 8 i 30 milionów bajtów. W tym samym czasie zorganizowano pokazy działania systemów teletransmisji danych w ZETO Wrocław, systemu POLRAX-2 oraz systemu WASC w Politechnice Wrocławskiej.

Wielodostępny system POLRAX-2 pracuje na ODRZE 1305 z dozbrojeniem sprzętu firmy ICL, ITT oraz RC. Oprogramowanie operacyjne stanowi system GEORGE 3. POLRAX-2 pozwala na łatwe włączanie do systemu oprogramowania firmowego serii 1300 oraz programów poszczególnych użytkowników. Ze względu na sposób komunikacji dopuszcza się lokalną oraz zdalną pracę wsadową, a także pracę interakcyjną i konwersacyjną. System WASC dopuszcza następujące rodzaje współpracy z komputerem:

— przetwarzanie partiowe w centrum obliczeniowym
— przetwarzanie zdalne z końcówek abonenckich sterowane systemem MINIMOP. MINIMOP przewiduje — poza pracą interakcyjną — również pracę w trybie konwersacyjnym przy wykorzystaniu języka JEAN. MINIMOP dopuszcza uruchamianie programów w językach PLAN 3, FORTRAN i ALGOL. Oprogramowanie systemu WASC znajduje zastosowanie w:
— dydaktyce
— obliczeniach numerycznych
— informacji naukowo-technicznej i ekonomicznej
— pracach projektowych
— automatyzacji pomiarów
— zarządzaniu szkołą wyższą.

Zainteresowanie pokazami było bardzo duże, szkoda tylko, że odbywały się one równolegle.

Należy wspomnieć, że w trakcie konferencji otwarto wystawę książki naukowej poświęconej zagadnieniom informatyki.

Zenon Grodzki
Andrzej Piaskowski

*) Zgłoszono udział 624 osób.

Informatyka w budownictwie lubelskim

W październiku ubiegłego roku Terenowy Ośrodek Koordynacji Branżowej w Projektowaniu Budownictwa dla województwa lubelskiego, Biuro Wiodące w Ośrodku — BPBBO „Miastoprojekt — Lublin" zorganizowało Sympozjum na temat: „Zastosowanie minikomputerów do obliczeń naukowych, automatyzacji procesów projektowania i przetwarzania w przemyśle budowlanym" z udziałem prelegenta z Biura Projektów Budownictwa Ogólnego w Warszawie mgr inż. Wiesława Płocharza i przedstawiciela firmy WANG z Wiednia.
Program Sympozjum obejmował:

— zapoznanie słuchaczy z uzyskanymi doświadczeniami w zakresie wykorzystania systemu minikomputera WANG do obliczeń naukowych, inżynierskich i przetwarzania danych
— demonstrację urządzeń i systemów minikomputera WANG
— wykonanie niektórych obliczeń dla uczestników sympozjum, dotyczących zagadnień, nad którymi aktualnie pracują.

Sympozjum spotkało się z dużym zainteresowaniem ze strony pracowników biur projektowych, o czym świadczy obecność 80 osób przy 40 zaproszonych.
Aktualnie Terenowy Ośrodek Koordynacji Branżowej w Projektowaniu Budownictwa przygotowuje Sympozjum na temat: „Zastosowania kalkulatorów programowanych typu COMPUCORP do obliczeń inżynierskich i numerycznych w budownictwie", na którym zostanie zaprezentowany pełny zestaw urządzeń uzupełniających.
Lubelskie Zjednoczenie Budownictwa przygotowywuje się do szerokiego wprowadzenia informatyki do zarządzania produkcją budowlaną i korzystania z usług organizowanego Ośrodka Informatyki Budownictwa ETOB w Lublinie. Ustlono, że przedsiębiorstwem pilotującym wdrażanie będzie Lubelskie Przedsiębiorstwo Budownictwa Miejskiego. W związku z tym przeszkolono prawie wszystkich pracowników umysłowych tj. około 200 osób w zakresie podstawowych wiadomości z informatyki. Powołana komórka ETO LZB zorganizowała wspólnie z Ośrodkiem Szkolenia i Doskonalenia Kursowego LZB kurs dla kadry kierowniczej w zakresie wybranych zagadnień z informatyki, w którym uczestniczyło ponad 50 osób z przedsiębiorstw budowlanych. Zorganizowano też kurs dla koordynatorów systemów, w którym uczestniczyły osoby wchodzące w skład komórek ETO powołanych w przedsiębiorstwach Zjednoczenia. W br. komórka ETO LZB planuje zorganizowanie jeszcze dwóch kursów dla kadry kierowniczej i dwóch — dla koordynatorów systemów. Obecnie prace komórki ETO LZB koncentrują się na sprawach związanych z organizacją Ośrodka Informatyki ETOB i przygotowaniem użytkowników do stosowania ETO. Całością prac kieruje Główny Specjalista d/s Informatyki LZB — mgr Ryszard Burek.

Planowanym terminem przekazania Ośrodka Informatyki ETOB (wyposażonego w zestaw ODRA 1305) do eksploatacji jest II kwartał 1976 r. (R.B.)

Informatyka w Przemyśle Siarkowym

Z początkiem 1975 roku w Ośrodku Badawczo-Rozwojowym Przemysłu Siarkowego SIARKOPOL w Machowie k/Tarnobrzega utworzono pracownię automatyki kompleksowej. Głównym jej zadaniem jest wdrażanie maszyn cyfrowych do sterowania procesów technologicznych i operatywnego kierowania produkcją w kombinacie wydobywczym i przetwórczym siarki. Fakt, iż na stanowisko kierownika pracowni został powołany mgr inż. Kazimierz Duliński, główny inżynier do spraw automatyki i pomiarów w kombinacie, wydaje się nie tylko kwitować zaangażowanie i walory osobiste człowieka, lecz przede wszystkim świadczyć o chęci ścisłego powiązania działalności nowej placówki z potrzebami i realiami produkcji przemysłowej. Tym większa odpowiedzialność staje się udziałem konsultanta naukowego przedsięwzięcia. Roli tej podjęła się Politechnika Rzeszowska, a dokładniej — kierowany przez docenta Jana Gruszeckiego Zakład Informatyki i Automatyki Lotniczej.

Przedstawieniu zamierzeń informatycznych SIARKOPOLU było poświęcone dwudniowe sympozjum zorganizowane pod koniec lutego w Baranowie Sandomierskim. Udział w sympozjum wzięli przedstawiciele tych krajowych placówek naukowo-badawczych i badawczo-projektowych, które z SIARKOPOLEM już współpracują lub do takiej współpracy są brane pod uwagę. Zaproszono też przedstawicieli największego producenta minikomputerów — amerykańskiej firmy DIGITAL EQUIPMENT CORPORATION oraz przedstawicieli włoskiego państwowego koncernu ENI (ENTE NAZIONALE IDROCARBURI) od lat współpracującego z polskim przemysłem chemicznym — ostatnio w ramach umowy o współpracy zawartej z Ministerstwem Przemysłu Chemicznego. ENI był w Baranowie reprezentowany przez dwie firmy: konsultacyjno-softwareową TEMA z Bolonii oraz NUOVO PIGNONE — oferującą komputerowe systemy sterowania i transmisji danych (dostawca takich systemów m.in. dla Huty im. Lenina oraz dla toruńskiej ELANY). Ze względu na to, że NUOVO PIGNONE stosuje w swych systemach jednostki centralne INTERDATA, byli też obecni przedstawiciele tej firmy.
SIARKOPOL zamierza w najbliższym pięcioleciu zrealizować trzy duże systemy sterowania i operatywnego kierowania produkcją dla następujących obiektów przemysłowych:
— kopalni siarki w Machowie (wydobycie metodą odkrywkową)
— kopalni siarki w Jeziórku (wydobycie metodą podziemnego wytapiania)
— zakładów chemicznych w Machowie (m.in. wytwórnie kwasu siarkowego, kwasu fosforowego, superfosfatów).

Przewiduje się, że każdy z systemów będzie operował dużym zbiorem danych. Jest to bodaj jedyne makropodobieństwo. Poza tym już teraz rysują się istotne różnice między systemami. W pierwszym z nich na czoło wybijają się problemy transmisji danych, w drugim nie bardzo wiadomo jak się zabrać do opracowania modeli matematycznych procesów wydobywczych, w trzecim wreszcie dominują problemy automatyki przemysłowej. Zróżnicowanie to poważnie ograniczy możliwości unifikacji prac badawczych, projektowych i wdrożeniowych. Utrudni koordynację prac. Ograniczy też zapewne wykorzystywanie czwartego, „laboratoryjnego" systemu jaki SIARKOPOL zamierza kupić, głównie do symulowania trzech systemów przemysłowych. Trudne więc zadania stoją przed młodym lecz ambitnym zespołem. Nasuwa się tu jednak refleksja, że w Baranowie czegoś brakowało. Za mało było wątpliwości, za mało otwartych czy wręcz nie rozpoznanych problemów. Za mało zainteresowania poszukiwaniami tych problemów. Odnosiło się wrażenie, że zbudowanie tych systemów to problem przede wszystkim techniczny, sprzętowy. A przecież coś musi być w tym, że tak naprawdę to systemy tego rodzaju nie wyszły jeszcze ze stadium eksperymentowania. Także w krajach technicznie rozwiniętych, gdzie bariery sprzętowe (w naszym tych barier rozumieniu) praktycznie nie istnieją.
Postęp jest wielkością wektorową, nieskalarną. Ważnym jest nie tylko intensywność pracy i inwestowania, lecz także wybrany kierunek działań. Liczy się bowiem tylko składowa wektora działań leżąca na kierunku naprawdę właściwym, choć niemożliwym do wyznaczenia a priori. Zespołowi z Machowa warto życzyć dobrej składowej.

(wład)

Przyszły system IBM

„FUTURE SYSTEM" jest roboczą nazwą oczekiwanej za dwa lub trzy lata nowej rodziny maszyn IBM, które zastąpi wysłużoną już rodzinę IBM 370. Szczególnie interesujące wydają się być zamierzenia firmy IBM w chwili, kiedy jesienią 1974 r. firma ta stwierdziła, że przyszłość informatyki opierać się będzie na „zdecentralizowanej inteligencji" a nie na „dystrybuowanym przetwarzaniu", dla którego charakterystycznym jest koncepcja maszyn IBM 360, 370 i CDC. Przedstawiciele firmy IBM stwierdzili wówczas, że zapotrzebowanie na maszyny średnie wygasa. Ponadto IBM wiedząc o podjętej produkcji przez firmę UNIDATA (SIMENS, CII, PHILIPIS) linii X opartej na koncepcji IBM 370 oraz produkcji Jednolitego Systemu Krajów RWPG. W warunkach narastającej konkurencji niezależnych producentów poszczególnych urządzeń preryferyjnych wymiennych z jednostkami centralnymi IBM, musi zaoferować prawdopodobnie zupełnie coś nowego w dotychczasowej koncepcji architektury maszyn cyfrowych. Pierwszym symptomem nowej serii jest minikomputer S/32, który wbrew dotychczasowej koncepcji IBM-u ma wbudowane w jeden blok wszystkie podstawowe elementy funkcjonalne systemu. Oczywiście jest to wszystko możliwe w przypadku małych zestawów komputerowych i zaawansowanej technologii miniaturyzacji.
Koncepcja ta uniemożliwia zakup urzadzeń zewnętrznych od innych niż IBM dostawców. Nawiasem mówiąc, IBM zatrudnia 4.5 tys. sprzedawców mini SYSTEM/7 i SYSTEM/3. Spośród nich wyłoniono grupe ok. 2000 osób do sprzedaży SYSTEMU/32. Przy założeniu, że jeden sprzedawca sprzeda trzy maszyny w ciągu dwóch lat będzie pracowało u użytkowników ok. 12 tys. egzemplarzy tego typu maszyn. Dzieki temu SYSTEM 3 stanie się najliczniejszą rodziną maszyn jakie kiedykolwiek wyprodukowano. (MAT)

Nowy minikomputer IBM

Po długim wyczekiwaniu IBM ogłosiło z początkiem stycznia 1975 roku sprzedaż następnego minikomputera — SYSTEM/32, który jest najmniejszym modelem w SYSTEMIE 3. Minikomputer ten jest właściwie inteligentnym urządzeniem końcowym, które może być albo urządzeniem samodzielnie liczącym, albo też końcówką podłączoną do większego systemu obliczeniowego. Miesięczna opłata dzierżawna w USA wynosi od 770 dol. do 1181 dol., co odpowiada prawdopodobnie cenie zakupu około 40 tys. dol.
SYSTEM/32 jest konkurencyjny dla rejestratorów danych takich firm jak BURROUGHS, NCR, DIGITAL EQUIPMENT, SINGER, LITTON, QUANTEL, NIXDORF i PHILIPS. Szacuje się, że rynek amerykański ma duże zapotrzebowanie na tego typu urządzenia i możliwości sprzedaży ocenia się na 500 tys. jednostek.
IBM sugeruje, że minikomputer jest przeznaczony dla użytkowników, którzy mają obroty handlowe od 1 — 10 mln. dol. i zatrudniają od 12 — do 250 pracowników. IBM określa SYSTEM/32 jako „najłatwiejszy, najmniejszy i najtańszy" komputer uniwersalny, jaki kiedykolwiek został wyprodukowany przez IBM.
Wraz ze sprzętem zapewniona jest dostawa pięciu różnych typowych pakietów oprogramowania. Oprogramowanie przeznaczone jest dla użytkowników, którzy nigdy nie korzystali z elektronicznej techniki obliczeniowej i nie zatrudniaja zawodowych informatyków. Obejmuje ono takie dziedziny jak: budownictwo, handel hurtowy żywności, papieru i urządzeń biurowych, małe szpitale, organizacje i stowarzyszenia (ewidencja członków). Oprócz pakietów zastosowań dostarczone zostaną programy sortowania, gospodarki kartotekami i rejestrowania danych.
Jako język programowania stosowany jest RPG II.
Koszt wykorzystywania poszczególnych pakietów oprogramowania waha się od 2—3 tys. dol. opłaty początkowej oraz od 60—140 dol. opłaty miesięcznej. Oprogramowanie dla gospodarki kartotekami zawiera trzy podstawowe funkcje zarządzania bazą danych: tworzenie i utrzymywanie kartotek, kierowanie do kartotek pytań oraz drukowanie odpowiedzi.
SYSTEM/32 składa się z pamięci dyskowej, drukarki, monitora ekranowego i klawiatury zawartych w jednej obudowie wielkości biurka. Standardowy model zawiera 16 K. bajtów pamięci operacyjnej i może być rozszerzony do 32 K Pojemność dysków wynosi od 5 — — 9.1 mln. bajtów. Istnieje pięć wariantów szybkości drukarki: 40 i 80 znaków/s oraz 50, 100 i 155 wierszy/min. Cena zakupu najmniejszego zestawu z pamięcią operacyjną 16 KB, pamięcią dyskową 5MB i drukarką 40 zn./s. kosztuje 33.1 tys. dol. Zestaw największy — pamięć operacyjna 32 KB, pamięć dyskowa 9.1 MB oraz drukarka 155 wierszy/min. — kosztuje 44.000 dol. IBM oferuje serwis konserwacyjny po koszcie od 165—215 dolarów miesięcznie.
Dotychczas zainstalowano 30 tys. maszyn należących do tej rodziny. Szacuje się, że wraz z modelem S/ /32 w ciągu dwóch najbliższych lat — rodzina S/3 przekroczy 50 tys. zainstalowanych egzemplarzy, zmieniając całkowicie pogląd, że IBM nie zajmuje się produkcją minikomputerów.
Dzięki wyeliminowaniu programistów nowi użytkownicy znacznie szybciej i taniej będą mogli zinformatyzować takie podstawowe funkcje przetwarzania danych, jak: płace. księgowość itp. Przewidywaną chłonność rynku USA na minikomputery przedstawia poniższe zestawienie (wartość w mln. dol., w nawiasach liczba egzemplarzy w tysiącach):

	1974	1975	1976	19977	1978
elektroniczne maszyny księgujące	382	432	477	504	540
	(19,7)	(22,4)	(25,8)	(28,0)	(30,0)
systemy minikomputerowe	76	170	300	450	560
	(1,7)	(4,0)	(7,5)	(12,0)	(16,0)
komputery małe	661	613	560	512	464
	(8,0)	(7,5)	(7,0)	(6,4)	(5,8)
łącznie	1119	1215	1337	1466	1564
	(29,4)	(33,9)	(40,3)	(46,4)	(51,8)

Bułgaria importuje sprzęt z USA

Na początku stycznia 1975 r. bułgarski JZOTIMPEX podpisał kontrakt z kalifornijską firmą WANG CO o wartości 1,3 mln. dol. na dostawę taśm i dysków magnetycznych. W kontrakcie zapewniona jest także obsługa techniczna tych urządzeń. Dyski i taśmy magnetyczne przeznaczone są dla bułgarskiego minikomputera uniwersalnego JOST-3100. Dr Ben Wang — Prezydent kalifornijskiego przedsiębiorstwa — przypuszcza, że transakcja ta pociągnie za sobą budowę fabryki tych urządzeń w Bułgarii, a w takim przypadku strona bułgarska zapewnia sobie prawo sprzedaży wymienionego sprzętu na pozostałych rynkach Europy Wschodniej. (MAT)

Komputer zamiast sprzedawców

Amerykańskie Stowarzyszenie Kupców Krajowych przewiduje, że zautomatyzowanie punktów sprzedaży (koncepcja POS=point—of—sale) wyeliminuje od 25 do 30% pracowników zatrudnionych w sklepach i domach towarowych. Opierając się na uniwersalnym kodzie wyrobów wyeliminowane zostaną ceny z metek. Rozwiązaniu takiemu sprzeciwiają się władze stanu Meryland, które zamierzają uchwalić ustawę nie zezwalającą na stosowanie metek bez cen. (MAT)

Bank danych na minikomputerze

CINCOM SYSTEM — firma software'owa (CINCINNATI) przeprogramowała swój pakiet DBMS *(Data Base Management System)* zwany TOTAL z dużych maszyn typu IBM 360, 370, HONEYWELL--200, CDC CYBER 70 na minikomputer IBM SYSTEM/3.
Funkcjonujący bank danych na tym komputerze może współpracować w wersji transmisji danych z pakietem TOTAL eksploatowanym na dużych maszynach. (MAT)

Kolejne rewelacyjne mini

W tydzień po ogłoszeniu przez IBM rozpoczęcia sprzedaży minikomputera SYSTEM/32 — w styczniu 1975 firma DIGITAL EQUPMENT CORPORATION (DEC) ogłosiła sprzedaż trzech modeli minikomputerów po rewelacyjnej cenie. DATASYSTEM-310 ma mieć zastosowanie w przetwarzaniu danych, zawiera PAO o pojemności do 32 K, monitor ekranowy, podwójny dysk elastyczny *(floppy disc)* o pojemności 260 tys. znaków. Cena zestawu wynosi 12 tys. dol. Oprogramowanie zwane COS310 *(Commercial Operating System)* umożliwia programowanie w języku DIBOL *(Digital Business Oriented Lauguage)*. Następny model CMS/1 *(Computational Minikomputer System)* ma mieć zastosowanie w projektowaniu inżynierskim. W identycznym zestawie ma kosztować również 12 tys. dol., z tym, że oprogramowanie oparte jest o FORTRAN IV i rozszerzoną wersję języka BASIC. Trzecim modelem jest CLASSIC *(Classroom Interaction Computer)* wyceniony na 7,9 tys. dol. lub wydzierżawiony w cenie 1 dol. na studenta i godzinę, przeznaczony do zastosowań dydaktycznych. Zestaw jest w strukturze identyczny z modelami poprzednio wymienionymi, z tym że PAO ma mniejszą pojemność (16 K), a pamięć dyskowa tylko 128 K znaków. (MAT)

Japońskie komputery w Chinach

Japońska firma HITACHI wysłała delegację swoich ekspetrów do Chin w celu nawiązania współpracy. Wprawdzie Japończycy nie liczą w chwili obecnej na liczne zamówienia, ale przypuszczają, że aby zaplanować przyszły zbyt, konieczne są wstępne rozmowy rozpoznawcze z ekspertami chińskimi. Japończycy oferują maszyny HITACHI 8700, 8800 oraz w serii M model 180, który równolegle zaoferowano do sprzedaży w Ameryce. (MAT)

Japońskie komputery atakują rynek amerykański

Dziesięć lat temu nie istniał jeszcze przemysł komputerowy w Japonii. Po licznych zakupach licencji na produkcję komputerów, głównie z Ameryki i Francji, Japończycy są obecnie gotowi do eksportu swoich komputerów na rynek amerykański.
Firma HITACHI pertraktuje z amerykańskimi firmami CONTROL DATA i ITEL w sprawie wykorzystania systemu tych firm do sprzedaży japońskiej maszyny M--180. Maszyna ta zalicza się do komputerów dużych jest konkurencyjna w stosunku do IBM 370/168 i Japończycy wahają się jeszcze z zaoferowaniem jej sprzedaży. Rozważają oni możliwość zaoferowania w pierwszej kolejności komputerowych urządzeń oraz czytników dokumentów pisanych ręcznie.
Producenci amerykańscy natomiast obawiają się niekorzystnych skutków dla amerykańskiego przemysłu komputerowego, podobnych jakie powstały już w przemyśle motoryzacyjnym (niskie ceny wyrobów japońskich na rynku amerykańskim). (MAT)

Zwolnienia pracowników w HONEYWELL

HONEYWELL po wykupieniu w 1970 przedsiębiorstw komputerowych z koncernu GENERAL ELECTRIC napotyka trudności w zbycie blisko 20 modeli wzajemnie niewymiennych komputerów.
Jak było łatwe do przewidzenia, HONEYWELL w tej sytuacji musi zwolnić 4% swoich pracowników w fabrykach w Bostonie, Fenix i Santiago. Część pomieszczeń fabrycznych zgłoszona została do sprzedaży. Z informacji przemysłowych wynika, że technologia zaawansowanej w kwietniu 1974 r. nowej serii maszyn „60" nie wymaga tyle personelu, co produkowany dotychczas sprzęt. (MAT)

Informatyka w Brazylii

Brazylia jest atrakcyjnym rynkiem zbytu komputerów. Przemysł komputerowy w Brazylii wzrasta z roku na rok o 30%.

Wykorzystująca tanią siłę roboczą firma IBM uruchomiła w 1972 roku w Campinos fabrykę pamięci operacyjnych, monitorów ekranowych i pamięci taśmowych, a także zestawów SYSTEM 370 z przeznaczeniem na export. Podobnie firma BURROUGHS przenosi do Brazylii swoje fabryki z Detroit. Również angielska firma FERRANTI uruchamia produkcję komputera FM 1600 wspólnie z brazylijskim przedsiębiorstwem państwowym DIGIBRAS. Natomiast firma francuska LOGABAX zamierza uruchomić w Brazylii produkcję minikomputerów.

Brazylijski rynek użytkowników jest w 90% zdominowany przez firmy IBM i BURROUGHS. Pozostałe 10% rynku dzielą między sobą firmy UNIVAC, HONEYWELL-BULL, CII, TELEFUNKEN i OLIVETTI. Brazylijskie stowarzyszenie użytkowników komputerów (SVCESC) szacuje, że liczba instalacji komputerowych w Brazylii wynosi około 2500 egzemplarzy w tym ok. 500 minikomputerów. To samo stowarzyszenie szacuje, że do 1976 roku wartość instalacji komputerowych przekroczy 1 mld. dol. Brazylijczycy mają nadzieję, że w bardzo krótkim czasie wejdą do grona największych użytkowników komputerów w świecie. Ocenia się, że poziom zastosowań w Brazylii jest w porównaniu do USA opóźniony o ok. 4 lata. Obecnie w Brazylii jest instalowany każdego dnia 1 komputer. Rządowa Agencja Łączności uruchomiła już ogólnodostępną sieć transmisji danych EMBRATEL, która zapewnia przesyłanie danych z szybkością 9600 bodów. (MAT)

Następny proces przeciw IBM

Firma SANDERS ASSOCIATES zapowiedziała wytoczenie procesu sądowego firmie IBM z żądaniem odszkodowania w wysokości 250 mln dolarów za bezprawne — zdaniem tej firmy — zmienianie parametrów urządzeń synchronizujących działałanie jednostki centralnej z urządzeniami końcowymi.
W odpowiedzi na ten zarzut Prezes firmy IMB Frank T. Gary stwierdził, że oskarżenie to jest bardzo podobne do innych wysuwanych przeciw IBM. Wynika z nich — zauważa Frank Gary, że IBM powinien chronić interesy konkurentów, zamiast ulepszać swoje wyroby w interesie użytkowników. (MAT)

Informatyka w centrali handlu zagranicznego

Aktywny udział obrotu towarowego z zagranicą w dynamizowaniu wzrostu dochodu narodowego, zobowiązuje centrale handlu zagranicznego do doskonalenia organizacji obrotu; automatyzowania zarządzania oraz do przyspieszania obiegu informacji. W tym celu centrale sięgają po elektroniczne maszyny cyfrowe by z ich pomocą wdrażać odpowiednie systemy informatyczne.

Specyfika handlu zagranicznego wymaga rozpatrzenia takich systemów w dwu aspektach:

— jako narzędzia służącego do szybkiego wystawiania dokumentów źródłowych

— jako urządzenia służącego do ewidencji i przetwarzania informacji, które ułatwią realizację i kontrolę funkcji centrali.

W Resortowym Ośrodku Informatyki i Doradztwa Organizacyjnego ETOCHEM, podjęto m.in. prace nad systemem informatycznym dla Centrali Importowo-Eksportowej Chemikaliami CIECH, którego cele skupiają się wokół realizacji wymienionych zadań.

Wystawiane dokumentów źródłowych

Dla usprawnienia procesu wystawiania dokumentów źródłowych w Centrali CIECH dokonano zakupu mikrokomputerów PROGRAMMA/P-603 firmy OLIVETTI.

W pierwszej fazie wdrożeniu podlegać będzie wystawianie dokumentów statystycznych:

— pozwolenie wywozu (PW)
— pozwolenie przywozu (PP)
— faktura eksportowa (FE)
— faktura importowa (FI)

W Centrali CIECH codziennie wystawia się od kilkudziesięciu do kilkuset dokumentów. W Dziale Handlowym przygotowuje się Karty Informacyjne, które stanowią podstawę do wystawiania dokumentów przez mikrokomputer PROGRAMMA/P-603.

Wypełnianie Kart Informacyjnych jest czynnością eliminującą pracochłonny proces tradycyjnego wypisywania informacji i obliczanie wartości występujących w poszczególnych dokumentach, a zatem — mimo konieczności wpisywania informacji dodatkowych — oszczędza czas pracowników handlowych umożliwiając im koncentrację wysiłku na zadaniach będących rzeczywistym przedmiotem ich pracy.

Mikrokomputer PROGRAMMA/P-603 wyposażony jest w urządzenie peryferyjne pozwalające na jednoczesne z wystawianiem dokumentu emitowanie maszynowego nośnika informacji. Nośnikiem tym może być taśma papierowa lub kasetowa taśma magnetyczna CTU. Utworzony w ten sposób maszynowy nośnik informacji zawiera dane źródłowe dla Małego Systemu Informatycznego CIECH.

Mały System Informatyczny CIECH

Przewiduje się opracowanie dla Centrali CIECH systemu informatycznego CIECH-INFOSYSTEM, obejmującego następujące podsystemy

— Dokumenty Eksport/Import
— Koszty
— Rozliczenia — Finanse
— Rynki — Kontrahenci
— Towary — Produkcja

W chwili obecnej zajęto się opracowaniem podsystemu Dokumenty Eksport/ /Import, ograniczając się do czterech podstawowych dokumentów statystycznych (PW, PP, FI, FE), w oparciu o maszynowy nośnik informacji tworzony w Centrali CIECH przy wystawianiu tych dokumentów przez mikrokomputer.

Wydawnictwa

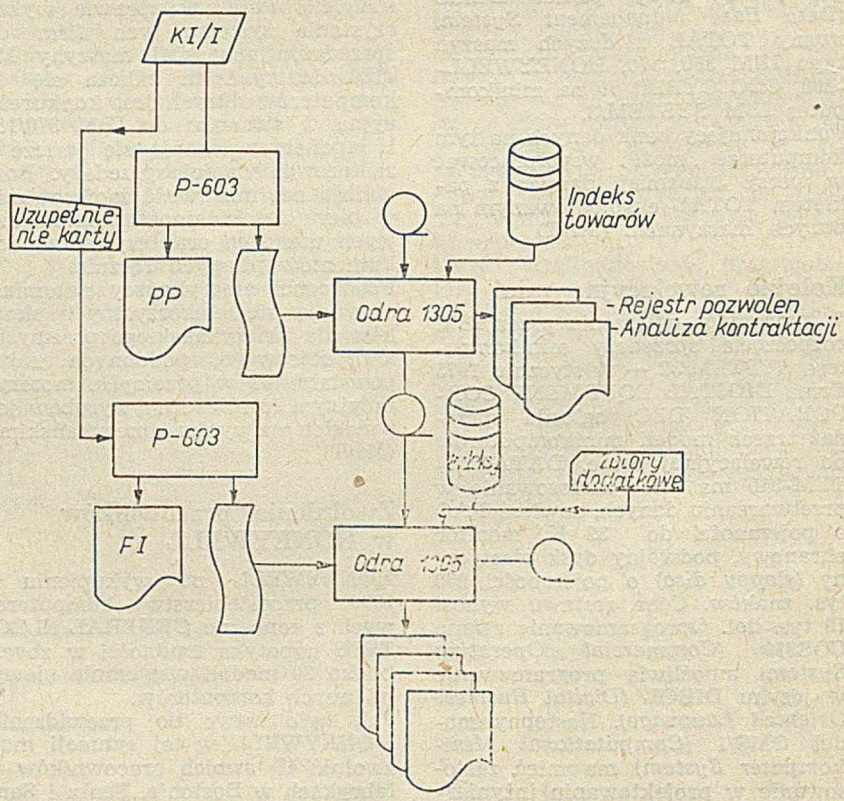
Zakres systemu jest ograniczony zawartością informacyjną dokumentów źródłowych, mimo to wachlarz emitowanych wydawnictw jest stosunkowo szeroki.

Wydawnictwa przeznaczone są dla biur branżowych oraz dla Centrali CIECH jako całości. Mają one charakter dostosowany do wymogów nowych, kompleksowych zasad ekonomiczno-finansowych zarządzania. Odpowiednie działy biur branżowych otrzymywać będą informacje niezbędne do wykonywania swoich zadań.

Dotyczy to szczególnie:

— planowania obrotu towarowego i korekty planów
— prowadzenia ewidencji i sprawozdawczości stanu kontraktacji i stanu realizacji eksportu i importu
— analizy cen krajowych i dewizowych oraz efektywności eksportu i importu
— przygotowania danych do analizy koniunktur na poszczególne towary
— opracowań różnych, dotyczących działalności biur dla potrzeb kierownictwa Centrali.

Poszczególne działy Centrali CIECH otrzymają informacje w takim samym zakresie, jednak o wyższym stopniu agregacji i w nieco innych układach. Ponadto otrzymają szereg informacji dodatkowych dotyczących następujących problemów:



— przygotowania materiałów do pertraktacji z klientami zagranicznymi na szczeblu dyrekcji naczelnej lub kilku biur branżowych

— bieżącej kontroli stanu realizacji umów handlowych, kontraktów i wykonania zadań planowych

— koordynacji kierunków działalności biur branżowych

— prowadzenia ewidencji obrotów wg poszczególnych rynków z uwzględnieniem najważniejszych partnerów i podstawowego asortymentu

— sporządzania meldunków o kontraktacji i wykonaniu obrotu towarowego z zagranicą

— sporządzania rozliczeń efektywności zrealizowanego eksportu i importu

— ewidencji i kontroli wystawianych pozwoleń przywozu i wywozu

— ewidencji i kontroli faktur eksportowych i importowych (statystycznych)

— planowania funduszu przedsiębiorstwa z podziałem na akwizycje, podróże i agentów

— analiz kierunków i rodzajów przewozu masy towarowej.

W miarę potrzeby istnieje możliwość dokonywania obliczeń tylko dla eksportu lub importu albo dla obu tych dziedzin jednocześnie. Jednorazowo wprowadzone informacje z dokumentów pozwolą na uzyskanie — w danym cyklu obliczeniowym — wydawnictw ze wszystkich lub tylko wybranych modułów.

W trakcie emitowania wydawnictwa na drukarce wierszowej dokonuje się zapisu informacji wynikowych na taśmie magnetycznej, celem zabezpieczenia przetwarzania przed konsekwencjami niesprawności drukarki lub uzyskania większej ilości egzemplarzy wydruku. Opracowano także programy parametryczne pozwalające na uzyskanie wyników w różnych stopniach agregacji i o różnych obszarach sumowań. System zaprojektowano w sposób umożliwiający przeprowadzenie obliczeń zarówno dla eksportu, jak i dla importu, przy wykorzystaniu tych samych modułów (por. rys. 1). Zastosowanie oddzielnych modułów dla eksportu i importu przewidziano jedynie dla przypadków znacznego zróżnicowania algorytmów przetwarzania.

W systemie przewiduje się korzystanie ze zbiorów dyskowych. Zbiory zawierające informacje niezbędne do kontroli danych źródłowych (symbole) i dla wydruków (nazwy) zorganizowane będą w systemie indeksowym.

Dzięki gromadzeniu i przechowywaniu wszystkich informacji wprowadzonych na podstawie dokumentów statystycznych istnieje możliwość rozbudowania systemu w dowolnych kierunkach, tak dla potrzeb CIECH, jak i dla innych jednostek organizacyjnych (MHZ, MP Chem, GUS, Zjednoczenia, Przedsiębiorstwa itp.).

Stan uruchomienia

W Centrali Importowo-Eksportowej Chemikaliami CIECH zainstalowano już kilka mikrokomputerów PROGRAMMA/P-603; aktualnie wdraża się już przewidzianą w systemie emisję dokumentów. Wdrożenie pierwszych modułów Małego Systemu Informatycznego CIECH przewiduje się w bieżącym roku.

Barbara Nowakowska
ETOCHEM

Warunki prenumeraty czasopism technicznych WCT NOT na rok 1975

1. Prenumerata dla zakładów pracy

Zamówienia na prenumeratę czasopism technicznych WCT NOT, na znormalizowanych i zaktualizowanych drukach, należy kierować do **Działu Zamówień i Rozliczeń Wydawnictw Czasopism Technicznych NOT, 00-048 Warszawa, ul. Mazowiecka 12, tel. 26-80-16.**

Przyjmowane i realizowane będą jedynie zamówienia na prenumeratę roczną. Przesyłając zamówienie należy jednocześnie wpłacić należność za roczną prenumeratę wymienionych w zamówieniu czasopism. Przesłanie zamówienia i dokonanie wpłaty powinno nastąpić w nieprzekraczalnym terminie do 31 października roku poprzedzającego prenumeratę. Wpłacać należy na konto **PKO I O/M Warszawa nr 1-9-121697 Wydawnictwa Czasopism Technicznych NOT.**

Do stałych odbiorców-prenumeratów czasopism technicznych NOT będą wysyłane druki zamówień wraz z cennikiem oraz informacją o sposobie ich wypełniania. Nowi prenumeratorzy mogą składać zamówienia na dowolnych formularzach bądź drukach, podając pełną nazwę zamawiającej instytucji (płatnika), pełną nazwę i adres (z kodem) odbiorcy, tytuły zamawianych czasopism, liczbę egzemplarzy. Zamówienie powinno być podpisane przez dyrektora i głównego księgowego oraz zawierać informację o dacie dokonania wpłaty za prenumeratę.

2. Prenumerata indywidualna

Prenumeratorzy indywidualni mogą zamawiać i opłacać prenumeratę roczną, półroczną i kwartalną czasopism WCT NOT w dowolnym Urzędzie Pocztowym za pomocą blankietu PKO; wpłacając należność na konto 1-9-121697 Wydawnictw Czasopism Technicznych NOT Warszawa 00-048 ul. Mazowiecka 12. Na odwrocie blankietu PKO (w miejscu na korespondencję) należy podać tytuły oraz liczbę egzemplarzy.

3. Prenumerata dla jednostek i instytucji wojskowych

Zamówienia na prenumeratę czasopism technicznych WCT NOT przyjmuje wyłącznie **RSW „Prasa-Książka-Ruch" Centralny Kolportaż Wojskowy, 00-950 Warszawa, ul. Grzybowska 77, skr. poczt. 2039, konto NBP VIII O/M Warszawa nr 1532-6/01-81116.** Warunkiem przyjęcia zamówienia do realizacji jest terminowe przesłanie go do CKW z jednoczesnym wpłaceniem należności. Termin przesłania zamówień i wpłacenie należności za prenumeratę zamówionych czasopism upływa z dniem 31.X. roku poprzedzającego prenumeratę

4. Prenumerata z rabatem

Członkom SNT NOT, nauczycielom, studentom, uczniom szkół technicznych przysługuje rabat 33% od cen normalnych, zgodnie z Uchwałą Zarządu Głównego NOT. Zamówienia na prenumeratę ulgową należy składać zgodnie z zasadami w punkcie 1 z jednoczesnym podaniem numeru legitymacji uprawniającej do zniżki.

Wszelkich dodatkowych informacji i wyjaśnień udziela Dział Prenumeraty WCT NOT, 00-048 Warszawa, ul. Mazowiecka 12, telefon 26-85-88.

Pragnąc zaspokoić życzenia licznych Czytelników, domagających się coraz częściej podjęcia przez nas prób porządkowania tak żywiołowo rozwijającej się terminologii informatycznej, rozpoczynamy od bieżącego numeru publikowanie stałej rubryki poświęconej tej problematyce. Przy okazji przypominamy, że w tej dziedzinie istnieje już w Polsce pięć oficjalnych dokumentów o charakterze norm państwowych (PN), a mianowicie:

- PN-71/T-01016 „Przetwarzanie danych i komputery. Podstawowe nazwy i określenia", obowiązująca od 1.07.1972 r., a obejmująca 253 terminy
- PN-71/T-01017 „Binarne elementy cyfrowe. Nazwy i określenia", obowiązująca od 1.07.72 r., a obejmująca 65 terminów
- PN-74/T-42100 „Karty dziurkowane 80-kolumnowe. Nazwy i określenia", stanowiąca nowelizację normy PN-66/M-42100 z 1966 roku, a obejmująca 31 terminów (obecnie w druku)
- PN-74/T-42102 „Taśmy dziurkowane. Nazwy i określenia", stanowiąca nowelizację normy PN-68/M-42102 z 1968 roku, a obejmująca 19 terminów (obecnie w druku)
- PN-74/T-42104 „Taśmy magnetyczne cyfrowe. Nazwy i określenia", stanowiąca nowelizację normy PN-68/M-42104 z 1968 roku, a obejmująca 47 terminów.

Tak więc aktualnie obowiązujące u nas w zakresie terminologii informatycznej dokumenty normalizacyjne obejmują łącznie zaledwie 415 pojęć.
Jest to oczywiście niewielki odsetek istniejącej już terminologii informatycznej, bowiem niektóre wydane w ub.r. słowniki informatyczne obejmowały już ponad 20 000 terminów.

Należy stwierdzić, że szczególnie trudne jest precyzowanie treści definicji odpowiadających określonym terminom. Wynika to z wyjątkowo szybkiego tempa rozwoju zarówno konstrukcji i technologii sprzętu, jak i technologii użytkowania środków informatyki. Nowe pojęcia i terminy tworzone w trakcie bieżącej działalności naukowej, projektowej, produkcyjnej lub eksploatacyjnej mają często charakter żargonowych terminów roboczych, jednak z braku innych, bardziej przemyślanych, rozpowszechniają się bardzo szybko, powodując często uzasadnione opory a nawet brak powszechnej akceptacji ze strony środowiska informatyków.

Z problemem tym w skali międzynarodowej boryka się już od przeszło 8 lat działający w ramach Komitetu Informatyki Międzynarodowej Organizacji Normalizacyjnej (ISO TC/97 „Computers and Information Processing") specjalny podkomitet słownictwa (97/SC 1 „Vocabulary"). Ostatnio obserwujemy pewien widoczny postęp w wynikach prac tego podkomitetu, który do końca ub. roku zdołał uzgodnić i opublikować już 9 projektów norm międzynarodowych DIS (Draft International Standard), obejmujących łącznie ok. 800 terminów z 9 różnych dziedzin problematyki informatycznej.
W roboczym opracowaniu wspomnianych dokumentów normalizacyjnych ISO uczestniczą bezpośrednio przedstawiciele krajów przodujących w rozwoju informatyki. Trwające od kilku lat opiniowanie i weryfikowanie kolejnych wersji dokumentów przez kilkanaście krajów członkowskich ISO (w tym również Polskę) sprawiło, że aktualna treść dokumentów, a zwłaszcza definicji w znacznej większości nie budzi już zastrzeżeń.
Ze względu na to, że spośród wielu różnych prób porządkowania słownictwa informatycznego dokumenty te stanowią aktualnie wzorzec najlepszy, zdecydowaliśmy się w naszej rubryce terminologicznej publikować w pierwszej kolejności wybrane terminy i ich definicje z niektórych ostatnio opracowanych dokumentów ISO. Znaczny udział problematyki programowania maszyn cyfrowych w treści zeszytów „INFORMATYKI" oraz największe potrzeby dydaktyczne w tej właśnie dziedzinie, skłoniły nas do rozpoczęcia cyklu od opracowania i opublikowania haseł wybranych z aktualnie opiniowanego projektu normy ISO/DIS 2382/VII „Digital Computer Programming". Obok propozycji polskiego terminu i adaptacji treści definicji zamieszczamy wzorem rozwiązania przyjętego w naszej normie PN-71/T-01016, również brzmienie terminu w języku angielskim. Uzupełnienie to, ze względu na powszechność (często może przesadną) użycia terminów angielskich, przyczynić się może do ułatwienia Czytelnikom oceny trafności proponowanego znormalizowanego terminu polskiego.

Warto podkreślić, że konieczność przyspieszenia prac nad tworzeniem prawidłowego i jednolitego słownictwa informatycznego wynika obecnie już nie tylko z postulatów publikacyjnych i dydaktycznych, ale również z potrzeb zautomatyzowanych systemów wyszukiwania informacji naukowo-technicznej.
Jak już wspomniano, doprowadzenie terminologii do powszechnej akceptacji społecznej jest sprawą trudną i długotrwałą — przykładem klasycznym była historia akceptacji u nas słowa „komputer". Dlatego też zwracamy się do jak najszerszego grona Czytelników o aktywne włączenie się do naszej akcji porządkowania terminologii i nadsyłanie wszelkich uwag i propozycji na temat publikowanych haseł i ich definicji. Przyczyni się to walnie poprzez powszechny sondaż opinii środowiska do przyspieszenia rozwoju i poprawy jakości polskiego słownictwa informatycznego.

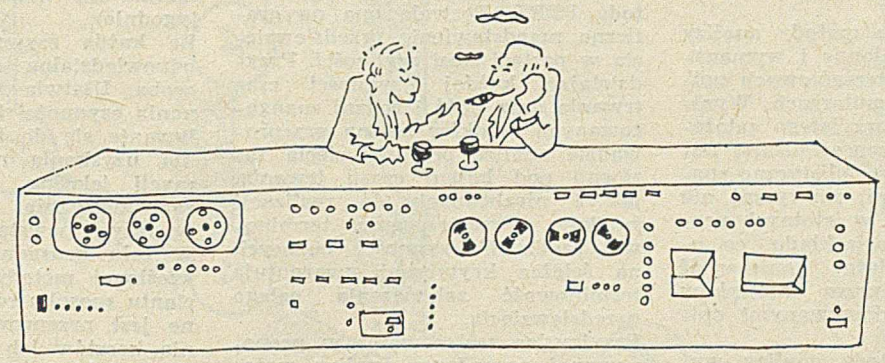
Władysław Klepacz

Programowanie komputerów cyfrowych (*digital computer programming*).
Terminy ogólne (*general terms*)

- **system programowania** *(programming system)* — jeden lub większa liczba języków programowania wraz z oprogramowaniem, umożliwiającym użycie tych języków na określonym sprzęcie do automatycznego przetwarzania danych
- **biblioteka programów, biblioteka** *(program library)* — zorganizowany zbiór programów komputerowych, udokumentowanych w sposób umożliwiający użytkowanie tych programów również przez osoby nie będące ich autorami.
- **program biblioteczny** *(library program)* — program komputerowy, zawarty lub pochodzący z biblioteki programów

- **procesor języka, procesor** *((langage) processor)* — program komputerowy, tłumaczący lub interpretujący albo też realizujący inne zadania, niezbędne do przetwarzania określonego języka programowania. Przykłady: procesor języka FORTRAN, procesor języka COBOL
- **preprocesor** *(preprocessor)* — program komputerowy, realizujący określone wstępne funkcje obliczeniowe lub organizacyjne
- **postprocesor** *(postprocessor)* — program komputerowy, realizujący określone końcowe funkcje obliczeniowe lub organizacyjne.
- **instrukcja** *(statement)* — wyrażenie znaczące w języku programowania, które opisuje lub określa operację i jest w kontekście tego języka kompletne
- **komentarz** *(comment, (computer program) annotation, remark, note)* — opis, odsyłacz lub wyjaśnienie, dodane lub wtrącone między instrukcje języka źródłowego, które nie wpływają na język wynikowy
- **rozkaz** *(instruction)* — wyrażenie znaczące w języku programowania, które określa jedną operację i podaje jej argumenty, jeśli one wystąpią
- **deklaracja** *(declaration, directive)* — wyrażenie znaczące w języku programowania, które wpływa na interpretację innych wyrażeń w tym języku
- **adres** *(address)* — znak lub grupa znaków, które podają rejestr, określoną część pamięci lub inne źródło względnie miejsce przeznaczenia danych
- **mikroprogram** *(microprogram)* — przechowywana w specjalnej pamięci sekwencja rozkazów elementarnych, odpowiadających jednej operacji komputera, której wykonanie inicjuje wprowadzenie rozkazu komputera do rejestru rozkazów
- **mikroprogramowanie** *(microprogramming)* — przygotowanie albo użycie mikroprogramów

TRYBUNA CZYTELNIKA



Redakcja INFORMATYKI

Jestem stałym czytelnikiem czasopisma INFORMATYKA, jeszcze od czasów, gdy nosiło ono nazwę MASZYNY MATEMATYCZNE i mogę z satysfakcją stwierdzić, że staje się ono coraz ciekawsze i coraz bliższe problemom, z jakimi styka się polski informatyk. Jako człowiek profesjonalnie związany z informatyką i długoletni pracownik ZETO w Łodzi odnotowuję ten fakt z wielkim zadowoleniem.

Jedną z ciekawszych i cennych form wydaje mi się podawanie syntetycznych wyciągów czy opracowań opartych o obcojęzyczną prasę fachową, w której napotyka się stale wiele materiału interesującego i polskiego czytelnika.

W chwili obecnej, gdy liczba różnorodnych ośrodków obliczeniowych gwałtownie wzrasta w naszym kraju, problemy ich organizacji i problemy organizacji służb informatycznych w przedsiębiorstwach — zaczynają stawać się tematyką dnia. Z ciekawym ujęciem tego zagadnienia spotkałem się w amerykańskim czasopiśmie DATAMATION z listopada 1974 roku z związku z czym pozwalam sobie opracowanie tego artykułu*) załączyć do mego listu, z myślą, iż może Redakcja zechce ten materiał wykorzystać.

Kreślę się z poważaniem
Ludwik Jerzy Rossowski

*) Materiał ten publikujemy w tym numerze na str. 30

Metody analizy kompleksowej systemu informacyjnego przedsiębiorstwa

Założenia organizacyjne do analizy kompleksowej

Wśród teoretyków informatyki wytworzyło się przekonanie o nieskuteczności a nawet niemożliwości stosowania analizy kompleksowej systemu informacyjnego w zakładach przemysłowych.

Taki sposób rozumowania wynikał najczęściej, z własnych niepowodzeń lub przesłanek teoretycznych akcentujących ogrom prac analitycznych. Sądzimy jednak, że zastosowanie komputera do tego typu prac powinno znacznie ułatwić rozwiązywanie trudności zbierania informacji zarówno w etapie ustalania faktów, jak i w etapie analizy faktów.

Proponowana metoda analizy kompleksowej jest przeznaczona dla przedsiębiorstw nawet o bardzo skomplikowanej strukturze organizacyjnej. Umożliwia ona przeprowadzenie analizy przez personel własny przedsiębiorstwa pod warunkiem jednak zastosowania komputera.

Obecnie stosowane metody analizy są bardzo pracochłonne i wymagają dokonywania szczegółowych opisów na wielu formularzach. Wynika to głównie z przyjętego założenia, że analiza przeprowadzana jest przez instytucje specjalistyczne spoza przedsiębiorstwa, zazwyczaj nie mające rozeznania w istotnych problemach badanego zakładu, co oczywiście powoduje konieczność posługiwania się przez analityków nadmiernie rozbudowywanymi opisami.

Proponowana metoda analizy wymaga dla opisu zadania jedynie dwóch formularzy formatu A4:

1. Identyfikacja zadania
2. Charakterystyka dokumentu (zbioru)

Oprócz tego prowadzony jest „Główny rejestr zadań", który ma na celu identyfikację poszczególnych zadań w postaci symboli numerycznych.

Niezbędne zestawienia dla analizy wykonuje komputer za pomocą trzech prostych programów.

Przepływ dokumentów w każdym analizowanym dziale przedsiębiorstwa wykonuje sie jeszcze na etapie ustalania faktów na formularzu „Graficzna interpretacja funkcji działu", który obejmuje wszystkie dokumenty wpływające, wypływające i przechowywane w badanym dziale. Na etapie tym w działach utrzymujących kontakty z otoczeniem przedsiębiorstwa (np. dział zaopatrzenia) należy zestawić sytuacje decyzyjne na formularzach „karta decyzji".

Szczegółowy opis tej metody zaprezentowany zostanie w następnym artykule.

Opracowanie planu prac

Jak już wspomniano analiza systemu informacyjnego wykonywana jest w dwóch zasadniczych etapach — ustalania i analizy faktów. Przeprowadzenie tak odpowiedzialnego i trudnego przedsięwzięcia powinno być poprzedzone właściwym przygotowaniem organizacyjnym.

Podstawową czynnością przygotowawczą zespołu analizy jest stworzenie prawidłowego planu realizacji przedsięwzięcia. Coraz powszechniej stosowaną techniką planowania skomplikowanych przedsięwzięć jest — jak wiadomo — metoda PERT. Pozwala ona na graficzne przedstawienie przedsięwzięcia w postaci sieci czynności. Przydzielając każdej czynności czas trwania oraz ilość i rodzaj zaangażowanych środków można przeprowadzić analizę przedsięwzięcia zarówno pod kątem czasu trwania, jak i niezbędnych do realizacji środków. Przestrzeganie terminowej realizacji czynności będących na ścieżce krytycznej gwarantuje terminowość zakończenia całego przedsięwzięcia.

Przyjmując zastosowanie w proponowanej procedurze analizy metody PERT, należy dla dokładnego opisu każdej czynności analizy, projektowania i wdrażania wypełnić dokument o nazwie „Karta Czynności" (tab. I) oraz przydzielić tym czynnościom kody pracowników odpowiedzialnych za ich realizację.

Dokument ten umożliwia przeprowadzanie planowania prac zarówno metodą manualna, jak i przy użyciu komputera. Obok znanych wielu zalet analizy maszynowej najbardziej cenna jest możliwość łatwego wykonywania w bardzo krótkim czasie aktualizacji sieci czynności, jakie wynikają z rzeczywistych warunków realizacji.

Przy opracowywaniu sieci należy zwrócić szczególną uwagę na przyjmowane długości czasu trwania poszczególnych czynności, jak również przyjąć określoną jednostkę czasu. Czynności trwające krócej umożliwiają wprawdzie łatwiejszą kontrolę realizacji prac (kontrola jest częsta i bardziej precyzyjna), ale równocześnie bardzo rozdrabniają całe przedsięwzięcie, a tym samym bardziej je komplikują. Natomiast czynności trwające dłużej utrudniają sprawdzenie stanu zaawansowania prac, lecz jednocześnie powodują, że sieć jest bardzo przejrzysta. Wskazane jest, aby przyjęte planowane odcinki czasowe były możliwie najmniej zróżnicowane pod względem długości.

W większości przedsiębiorstw działalność pracowników umysłowych planuje i rozlicza się kwartalnie. Plany te są następnie rozkładane na miesiące, a nawet dekady czy tygodnie. W związku z tym zaleca się, aby czynności w omawianej sieci działań obejmowały okres czasu nie mniejszy niż tydzień, a jednocześnie nie większy niż miesiąc (4 tygodnie).

Za każdą czynność powinna być odpowiedzialna jedna konkretna osoba. Ułatwia to kontrolę wykonania czynności i sprowadza odwoływanie się do tylko jednej osoby dla uzyskania niezbędnych informacji jak np. określenia stopnia zaawansowania prac, wyjaśnienia przyczyn powstania opóźnień itp.

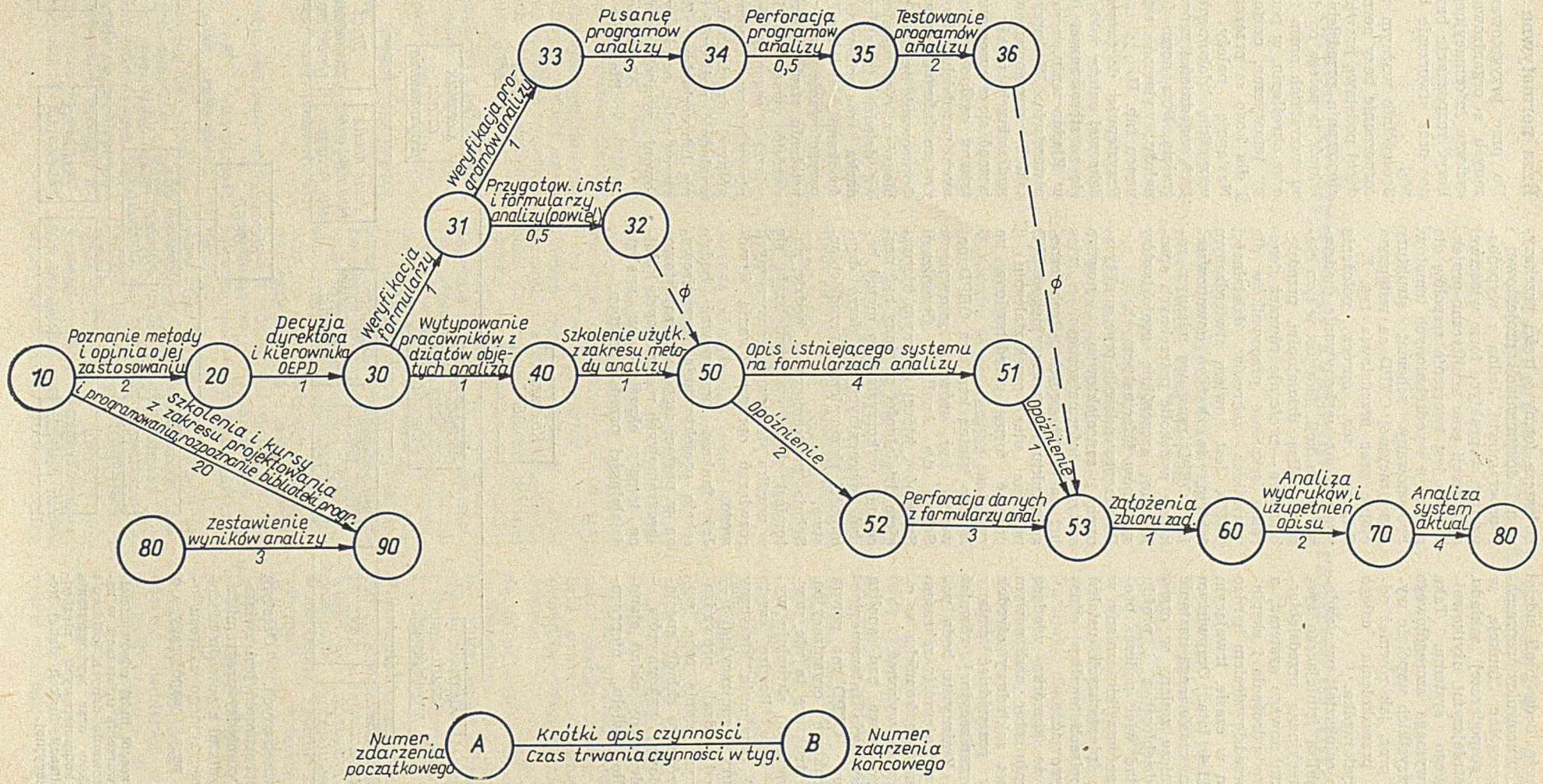
W celu uzasadnienia przyjęcia określonej metody analizy lub wariantu sposobu rozwiązania wskazane jest przeprowadzenie porównania kosztów ich realizacji. Porównania takiego można dokonać na podstawie kosztów skonkretyzowanych we wspomnianych już „KARTACH CZYNNOŚCI" (tab. I) dla całego przedsiewzięcia, etapu a nawet dla alternatywnych rozwiązań niektórych czynności.

Sieć czynności obrazującą przykład zastosowania metody maszynowej obrazuje rys. 1.

Organizacja prac

Proponowana maszynowa metoda analizy kompleksowej zakłada ścisłą współpracę ośrodka EPD ze wszystkimi funkcjonalnymi komórkami przedsiebiorstwa.

We wszystkich etapach prac związanych z wprowadzeniem systemu


10
Poznanie metody i opinia o jej zastosowaniu
2
20
Decyzja dyrektora i kierownika OEPD
1
30
Weryfikacja formularzy
1
Wytypowanie pracowników z działów objętych analizą
1
40
Szkolenie użytk. z zakresu metody analizy
1
50
Opis istniejącego systemu na formularzach analizy
4
51
Opóźnienie
1
Opóźnienie
2
Szkolenia i kursy z zakresu projektowania i programowania, rozpoznanie biblioteki progr.
20
80
Zestawienie wyników analizy
3
90
31
Przygotow. instr. i formularzy analizy (powiel.)
0,5
32
Weryfikacja programów analizy
1
33
Pisanie programów analizy
3
34
Perforacja programów analizy
0,5
35
Testowanie programów analizy
2
36
ϕ
ϕ
52
Perforacja danych z formularzy anal.
3
53
Założenia zbioru zad.
1
60
Analiza wydruków i uzupełnień opisu
2
70
Analiza system aktual.
4
80
Numer zdarzenia początkowego
A
Krótki opis czynności
Czas trwania czynności w tyg.
B
Numer zdarzenia końcowego

Rys. 1

informatycznego będą brali udział wytypowani pracownicy reprezentujący poszczególne funkcje w działalności gospodarczej zakładu. Będą oni późniejszymi użytkownikami [1]) nowego systemu. Taka współpraca wymaga szczegółowego określenia wszystkich wspólnie wykonywanych prac, jak również odpowiedniego przeszkolenia całego personelu użytkownika.

Jak wiadomo całokształt prac związanych z analizą istniejącego systemu informacyjnego przedsiębiorstwa, projektu nowego systemu, a następnie jego wdrożenia i eksploatacji, jest przedsięwzięciem bardzo rozciągniętym w czasie, trwającym co najmniej 4 lata. Oczywistym jest, że w tym czasie mogą nastąpić nawet zasadnicze zmiany w organizacji przedsiębiorstwa i profilu produkcji. Stąd więc podczas prowadzenia prac wszelkie zmiany w stosunku do pierwotnych ustaleń należy w opisach analizy odpowiednio korygować.

Przyjmując, że komórką wiodącą w prowadzeniu prac nad nowym systemem jest zakładowy ośrodek EPD, należy przewidzieć zmiany struktury organizacyjnej również tej komórki. Zmiany te będą podyktowane rodzajem i zakresem prac jakie musi ta komórka w określonym etapie wykonać.

Rozpoczynając prace nad analizą aktualnego systemu informacyjnego, zakładowy CEPD ma pewną ustaloną strukturę organizacyjną. W praktyce struktura ta nie zawsze dostosowana do aktualnie wykonywanych prac. Bardzo często ośrodki EPD przyjmują klasyczny układ struktury organizacyjnej i — niezależnie od rodzaju prowadzonych w danym etapie prac, wymagających różnego składu specjalizacji zawodowej wykonawców — utrzymują pierwotny układ stanowiący źródło wielu nieporozumień wewnętrznych. Częstym źródłem nieporozumień jest zwłaszcza konflikt pomiędzy programistami i projektantami z powodu niedostatecznie precyzyjnego zakresu obowiązków przy współpracy, a zwłaszcza formy i treści przekazywanej dokumentacji projektowej i i programowej. Jedną z wypróbowanych w praktyce form właściwej organizacji prac analityczno-projektowych jest tworzenie zespołów dla realizacji wyodrębnionych tematów. Proponowany schemat struktury organizacyjnej OEPD na wstępnym etapie prac przedstawia rys. 2.

Schemat ten oprócz położenia akcentu na wykonywanie podstawowej funkcji, jaką jest w tym etapie analiza działalności przedsiębiorstwa — preferuje również generalne szkolenie pracowników dla wszystkich zakresów działalności OEPD. Jest bowiem stwierdzonym faktem, że każdy nowopowstający OEPD musi we własnym zakresie wyszkolić potrzebnych specjalistów, szczególnie projektantów systemów informatycznych.

Po przeprowadzeniu prac analitycznych i zaprojektowaniu nowego systemu informatycznego, a więc już na etapie projektu technicznego, struktura organizacyjnego OEPD powinna ulec przekształceniu i przyjąć formę przedstawioną na rys. 3.

Bardzo istotną komórką OEPD powinna być wówczas rozwojowa Sekcja d/s Analiz i Organizacji Systemów, której głównym kierunkiem działalności powinny być prace nad wykorzystaniem metod badań operacyjnych w sferze działalności ekonomicznej przedsiębiorstwa.

Sekcja programowania będzie natomiast stopniowo zmniejszała się w wyniku stosowania nowoczesnych rozwiązań oprogramowania firmowego.

Jej funkcje przejmować będzie sekcja projektowania zorganizowana w postaci zespołów projektowo-programowych realizujących kolejne tematy określone w analizie działalności przedsiębiorstwa.

Baza normatywna i indeksowa

Do prac przygotowawczych związanych z organizowaniem nowego systemu informacyjnego należy zaliczyć rozpoznanie bazy normatywnej oraz stworzenie bazy indeksowej.

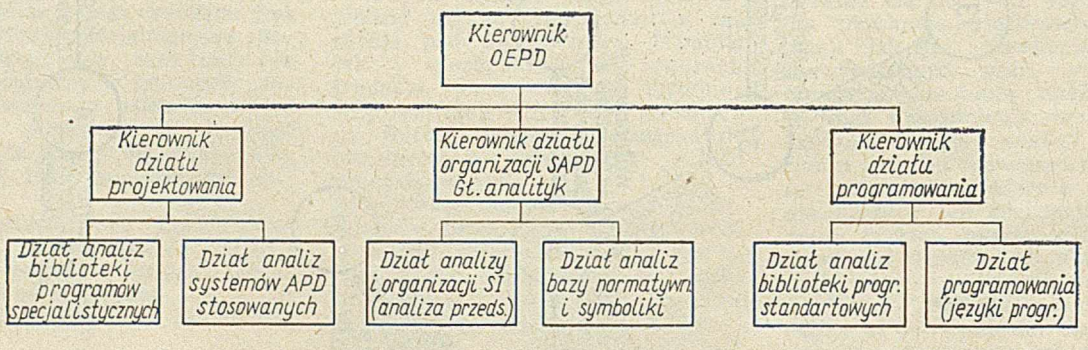
Szczególnie ważnym zadaniem jest rozpoznanie bazy normatywnej, będącej podstawą prawidłowego funkcjonowania przedsiębiorstwa. Działalność ta polega na dokonaniu spisu wszystkich norm i normatywów oraz określeniu, w podziale na grupy tematyczne, czego one dotyczą i dla kogo są przeznaczone.

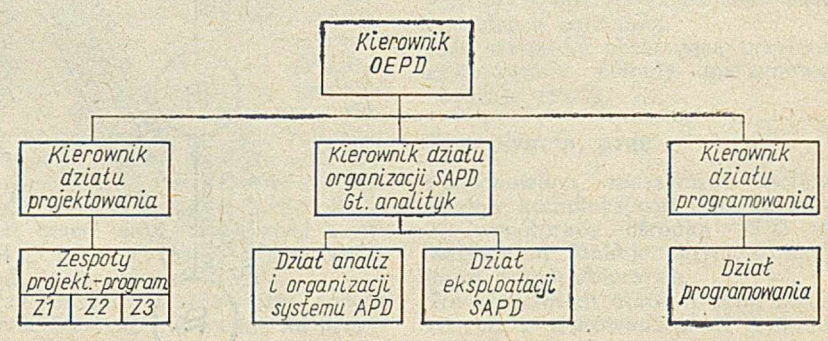
Następnie należy skompletować wszystkie dokumenty dotyczące tych zagadnień i zapoznać się z ich treścią. Spis norm i normatywów należy wykonać wg załączonego wzoru tab. II.

Przystępując do tworzenia bazy indeksowej należy dokładnie zapoznać się ze stosowanymi już symbolami w Zjednoczeniu, resorcie, czy ministerstwie. W przypadku braku takich opracowań w instytucjach nadrzędnych należy najpierw zapoznać się z wzorcowymi systematykami ogólnokrajowymi lub innych przedsiębiorstw realizujących system informatyczny.

Opracowanie symboli powinno przebiegać z góry ustalonym podziale wg dziedzin tematycznych (gospodarka materiałowa, gospodarka środkami trwałymi, techniczne przygotowanie produkcji itp.). Przy ustalaniu struktury symboli należy zwrócić uwagę na istniejące powiązanie organizacyjne oraz przepływy informacji pomiędzy poszczególnymi dziedzinami, a także uwzględnić możliwość ich uzupełniania lub rozbudowy. Ze względu na szereg zalet, należy preferować stosowanie symboli numerycznych.

Aleksander Adamowicz
Stanisław Szklarzewicz





[1]) Osoba lub zespół ludzi odpowiedzialnych za przetwarzanie informacji w zakresie przygotowania danych, odbioru i kontroli wyników oraz wykorzystania wydawnictw komputerowych niezbędnych dla realizacji funkcji danej komórki organizacyjnej.

Metody oceny efektywności systemów zarządzania bazą danych

Pytanie, jakie efekty przyniosło użytkownikowi zastosowanie systemu informatycznego sprowadza się często do odpowiedzenia sobie, w jakim stopniu system przyczynił się do osiągnięcia określonych przez użytkownika celów. Nie analizuje się zaś zależności istniejącej między wysokością tego stopnia, a sprawnością i wydajnością systemu wynikającą z jego rozwiązań metodycznych i projektowych. Tego rodzaju podejście kryje w sobie niebezpieczeństwo, że twórcy i projektanci systemu ograniczą się do osiągnięcia przez system poziomu skuteczności i wydajności mierzonego wyłącznie wygenerowaniem pożądanego i zdefiniowanego przez użytkownika „wyjścia" systemowego. Tymczasem niewłaściwie przeprowadzona ocena powoduje często obranie nieodpowiedniej metodyki projektowania i realizacji konkretnego już systemu. A przecież potencjalni użytkownicy systemów informatycznych mają swoje konkretne powody, dla których chcą u siebie wdrażać takie, a nie inne systemy o cechach z ich punktu widzenia najbardziej pożądanych. Przykładem mogą być odpowiedzi użytkowników na ankietę brytyjskiego National Computing Centre.

Podali oni następujące podstawowe powody, dla których podjęli starania w kierunku instalacji systemów informatycznych (typu systemów zarządzania bazą danych):

- *możliwość szybszej i tańszej realizacji zastosowania systemu*
- *łatwiejsza eksploatacja systemu*
- *niezależność danych ułatwiająca rozwój systemu*
- *redukcję nadmiaru informacji*
- *uzyskanie większej adekwatności informacji*
- *poprawę dostępności informacji*
- *większą szybkość i elastyczność w spełnianiu żądań użytkownika.*

Wymienione powody świadczą wyraźnie o dużych nadziejach, jakie pokłada się w sprawności systemów zarządzania bazą danych. Tym ostrzej więc rysuje się problem wypracowania właściwych metod i aparatu do kontrolowania oraz pomiarów skuteczności i wydajności tych systemów.

Badanie wydajności systemów zarządzania bazą danych

Obecnie można mówić o stosowaniu kilku metod badania wydajności systemu informatycznego. Metody te szeroko już opisywano w literaturze fachowej.
Są to między innymi:

- występna analiza *(primarily analysis)*
- symulacja *(simulation)*
- badania eksploatacyjne *(monitoring)*
- przebiegi porównawcze *(benchmarketing).*

Wymienione metody (poza symulacją) znajdują zastosowanie głównie w porównawczej ocenie istniejących już systemów i najczęściej sprowadzają się do analizy aspektu sprzętowego i oprogramowania podstawowego (systemu operacyjnego oraz oprogramowania użytkowego typu utility).

Metodę modelowania stosuje się do określenia przewidywanej wydajności systemu znajdującego się w fazie projektowania, wykorzystując jednocześnie dane na temat spodziewanego obciążenia i zachowania systemu uzyskane na podstawie statystyk eksploatacyjnych z systemu o podobnym typie.
Problem oceny wydajności i skuteczności systemów zarządzania bazą danych jest daleko bardziej złożony w przypadku konwencjonalnych systemów informatycznych. Zapewne z tego względu, jak również stosunkowo krótkiego okresu gromadzenia doświadczeń eksploatacyjnych z zastosowania tych systemów, brak jest jeszcze w literaturze fachowej wyczerpującego naświetlenia tych zagadnień (jakkolwiek ze względu na dużą złożoność systemów zarządzania bazą danych oraz ich dużą dynamikę użytkową, problemy badania i oceny sprawności i wydajności tych systemów znajdują się w centrum zainteresowania fachowców oraz użytkowników).
Analizując wydajność systemu zarządzania bazą danych, pracującego w trybie zdalnego dostępu *(on line)*, należy przede wszystkim rozpatrzyć elementy determinujące tę wydajność:

- sprzęt — podstawowa konfiguracja komputera oraz system komunikacyjny zdalnego dostępu (liczba terminali, typ i rodzaj terminali, linie telekomunikacyjne itp.)
- oprogramowanie (podstawowe oraz zastosowaniowe)
- użytkownicy systemu — w sensie współdziałania człowiek — maszyna *(man-machine interface).*

Pomijając elementy sprzętowe, najczęściej występujące w polu uwagi użytkownika, gdyż ich wydajność określona jest i uwarunkowana względnie stałymi parametrami techniczno-eksploatacyjnymi — głównym przedmiotem badań wydajności systemu stają się elementy oprogramowania oraz współdziałania człowiek — maszyna.

Badania eksplotacyjne

Badania eksploatacyjne są podstawową metodą stosowaną przy ocenie wydajności systemów zarządzania bazą danych. Rozumie się przez nie określone akcje ewidencyjno-pomiarowe podejmowane w czasie normalnej eksploatacji systemu. Akcje te wykonywane są za pomocą zespołu środków programowych (chociaż stosuje się również środki sprzętowe, np. specjalne urządzenia kontrolno-pomiarowe) związane głównie z oprogramowaniem obsługi funkcji zarządzania bazą danych i oprogramowaniem komunikacyjnym obsługi zadalnego dostępu.

Są to najczęściej odpowiednio zaprojektowane i oprogramowane procedury badań eksploatacyjnych, charakteryzujące się selektywnością i opcjonalnością stosowania. Umożliwiają one:

- dowolne ukierunkowanie procedur do przeprowadzenia określonego typu badań i testów pomiarowych
- swobodne, zgodnie z potrzebami, włączanie i wyłączanie procedur podczas eksploatacji systemu.

Procedury te można usystematyzować w trzy podstawowe grupy zależne od funkcji w procesie badań i pomiarów.

Procedury rejestracji określonych zdarzeń zaistniałych w czasie eksploatacji systemu. Za ich pomocą można rejestrować takie parametry jak: czas, źródło i przeznaczenie zdarzenia.

Procedury te obejmują następujący zakres zdarzeń:

— rejestrację włączania i wyłączania systemu
— rejestrację włączania i wyłączania terminali
— inicjację i zakończenie programu
— otrzymanie meldunku przez terminal
— niepoprawne zakończenie transakcji
— zawieszenie działania systemu z powodu usterek sprzętu lub oprogramowania
— próbę złamania ochrony dostępu do bazy danych.

Procedury gromadzenia statystyk i raportów. generowania Działają one w oparciu o zarejestrowane zdarzenia i umożliwiają gromadzenie szerokiego zakresu statystyk eksploatacyjnych dla każdego programu i transakcji.

Mogą to być:

— czas reakcji systemu *(systems response time)* mierzony jako czas między zakończeniem „wejścia" do systemu z terminalu, a zakończeniem wyjścia z systemu do kanału w celu przesłania informacji do terminalu
— czas reakcji użytkownika *(user ternaround time)* mierzony jako czas między zakończeniem „wyjścia" z systemu do kanału w celu przesłania informacji do terminalu, a zakończeniem „wejścia" do systemu z terminalu.
— czas cyklu współdziałania człowiek-maszyna mierzony jako suma czasów reakcji systemu i reakcji użytkownika
— czas pracy systemu
— czas pracy jednostki centralnej (CPU)
— liczba dyskowych operacji WE/WY
— liczba i nazwy wykorzystywanych zbiorów *(files)*
— nazwy i wartości danych indeksowych, wykorzystywanych w procesie selekcji.

Na podstawie wymienionych statystyk produkowane są określone raporty z częstotliwością tygodniową, miesięczną lub inną wynikającą z przebiegu badań.

Procedury statystyczno-analityczne. Służą one do wyliczania i analizowania pewnych średnich wartości elementów pomiarowych oraz określania trendów eksploatacyjnych systemu.

Wykorzystują one zazwyczaj dane statystyczne zgromadzone podczas stosunkowo długiego czasu trwania pomiarów eksploatacyjnych systemu np. kilku a nawet kilkunastu miesięcy, kilkudziesięciu tysięcy transakcji lub kilkuset programów.

Wygenerowane raporty analityczne będą charakteryzować zaistniałe w czasie pracy systemu warunki eksploatacyjne. Pozwolą na dokonanie oceny sprawności systemu poprzez analizę wielu podstawowych elementów.

Współdziałania człowiek—maszyna.

Mogą np. zawierać (podane dla określonych w/g typu grup programów zastosowaniowych) średnie wartości dotyczące:
— czasu reakcji systemu
— czasu reakcji użytkownika
— czasu pracy jednostki centralnej
— czasu trwania cyklu
— liczby cykli przypadających na sesję terminalową
— czasu trwania sesji
— procentowego obciążenia systemu przez dany typ programów itp.

- sprawności operacyjnej systemu przedstawionej poprzez podanie np. w odniesieniu do 1 godz. pracy systemu średnich wartości:
— czasu pracy jednostki centralnej
— czasu pracy całego systemu komputerowego
— czasu pracy terminali
— liczby zainicjowanych programów w ciągu jednej sesji terminalowej
— liczby transkacji itp.

- współdziałania między programi zastosowaniowymi, a systemem zarządzania bazą danych — szczególnie jego procedurami obsługi funkcji zarządzania bazą danych. Podaje się np. dla typów transakcji średnie wartości charakteryzujące:
— czas realizacji odwołania do procedur obsługi funkcji zarządzania bazą danych
— czas pracy jednostki centralnej dla realizacji odwołania
— liczbę dyskowych operacji WE/WY dla realizacji odwołania
— procentowe obciążenie procedur obsługi funkcji zarządzania bazą danych przez dany typ transakcji itd.

Scharakteryzowany powyżej aparat badań i pomiarów eksploatacyjnych pozwala przeanalizować wydajność systemu zarządzania bazą danych oraz zapewnia użytkownikowi inne wypływające stąd korzyści, z których najważniejszymi są:

- możliwość weryfikacji metod i rozwiązań projektowych przyjętych w systemie w szczególności dla:
— opisu logicznych elementów struktury bazy danych
— zdefiniowania powiązań elementów struktury bazy danych
— określenia zasad fizycznych — odwzorowania logicznej struktury bazy danych
— organizacji procedur awaryjnych
— zdefiniowania zasad ochrony dostępu do bazy danych i bezpieczeństwa danych
— określenia parametrów oceny efektywności eksploatacyjnej systemu

- zwiększenie skuteczności oraz możliwości operacyjnych systemu

- poprawa niezawodności działania systemu

- udoskonalenie sterowania bieżącą eksploatacją systemu.

Konkretne wyniki z badań i pomiarów eksploatacyjnych obrazują np. dane z opracowania J. D. Krinos'a dotyczące systemu United Aircraft Information Management System (UAIMS) *).

Wyniki pomiarów eksploatacyjnych na przykładzie systemu UAIMS UAIMS jest systemem pracującym równolegle w trybie lokalnym partiowym *(batch processing)* oraz w trybie zdalnego dostępu *(on line)* dostarczającym różnym grupom użytkowników z United Aircraft Corporation (UAC) potrzebną moc obliczeniową w zakresie „konwencjonalnego" przewarzania informacji oraz systemu bazy danych. UAIMS zaprojektowano w Laboratoriach Badawczych firmy United Aircraft Co i wdrożono do eksploatacji z początkiem 1970 r. na komputerze IB 360/50 z pamięcią 512 Kb oraz 8 jednostkami pamięci dyskowej IBM 2314 pod kontrolą systemu operacyjnego OS/MUT. UAIMS składa się z:

— systemu sterowania teleprzetwarzaniem *(teleprocessing executive* — TPE),
— systemu zarządzania bazą danych *(data base management system* — DMS),

— programów zastosowaniowych — ogólnego przeznaczenia i specjalistycznych.

Tabela I zawiera średnie wartości uzyskane w ramach pomiarów systemu UAIMS przeprowadzonych w ciągu 6 miesięcy przy łącznym zużyciu 700 godz. czasu komputera na przetwarzanie programów.

Wartości średnie podane zostały w odniesieniu do grup programów „konwencjonalnych", działających poza systemem zarządzania bazą danych (DMS), w ramach DMS i specjalnych, w ramach DMS, wykonujących allokację produkcji w drodze selekcji opartej o procedury iteracyjne.

Tabela II przedstawia średnie wartości w odniesieniu do 1 godz. pracy systemu UAIMS wynikające z pomiarów wykonanych w ramach codziennych czterech godzin pracy systemu w okresie 6 miesięcy Tabela III zawiera średnie wyniki uzyskane z pomiarów 70 tys. transakcji w ramach DMS w okresie jednego roku.

Wnioski

- Przedstawione korzyści ze stosowania badań i pomiarów eksploatacyjnych dostatecznie argumentują potrzebę stworzenia takiego aparatu programowego do oceny efektywności systemów zarządzania bazą danych.

- Procedury badań i pomiarów eksploatacyjnych w systemach zarządzania bazą danych stanowią podstawowe źródło informacji dla administratora bazy danych, umożliwiając mu bieżącą kontrolę eksploatacji bazy danych.

---

* Informatyczny System Zarządzania Koncernu Przemysłu Lotniczego w USA.

Tabela I. Charakterystyka programów

Charakterystyka Programów / Wartości Średnie	Programy poza DMS	Programy w ramach DMS		Razem Programy
		ogólne	specjalne	
Czas jednostki centralnej	0,2	1,6	3,7	1,3
Czas poza jednostką centralną	3,1	5,3	29,1	6,4
Czas reakcji systemu	3,3	6,9	32,8	7,7
Czas reakcji użytkownika	14,0	17,3	26,3	17,9
Czas cyklu	17,3	24,2	59,1	25.6
Ilość cykli/sesję	31	37	8	26
Czas sesji/min.	8,9	14,9	7,8	10,9
Obciążenie	33%	50%	17%	100%

Tabela II. Charakterystyki operacyjne

Wyszczególnienie		Wartość śr.
Praca jednostki centralnej		3,9 min.
Praca systemu komputerowego		23,3 min.
Praca terminali		1,4 godz.
Liczba uruchomionych programów		7
Liczba meldunków ze wszystkich terminali		182
Liczba transakcji w ramach DMS		73
Wykorzystanie DMS	Czas pracy DMS	7,4 min
	Czas pracy jednostki centralnej	2,2 min

Tabela III Charakterystyka transakcji DMS.

Typ transakcji / Wartości średnie	Funkcje sterujące		Wyszukiwanie	Umieszczanie danych w bazie	Aktualizacja	Transakcje razem
	różne	otwarcia zbiorów (files)				
Czas wykonania	1,3	3,6	5,9	6,1	13,2	5,6
Czas pracy jednostki centralnej	0,3	0,7	2,3	1,0	3,8	1,8
Liczba operacji dyskowych (WE/WY)	3,3	6,5	27,8	38,3	62,9	23,7
Obciążenie	5,4%	27,7%	55,3%	3,7%	7,9%	100%

● Projektowanie procedur badań i pomiarów eksploatacyjnych powinno odbywać się równolegle z projektowaniem systemu zarządzania bazą danych.

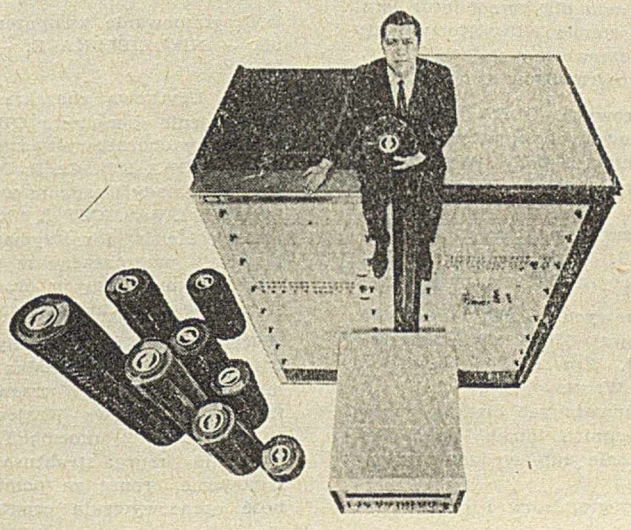
● Z uwagi na pogorszenie ogólnej wydajności systemu zarządzania bazą danych podczas prowadzenia badań eksploatacyjnych, procedury pomiarowe powinna cechować selektywność i opcjonalność stosowania, umożliwiająca ukierunkowanie ich na określone cele analiz oraz swobodne włączanie i wyłączanie w trakcie eksploatacji systemu.

● Należy pamiętać, że uzyskane wyniki z badań eksploatacyjnych zależą również w znacznym stopniu od parametrów techniczno-eksploatacyjnych zastosowanego sprzętu. Jest to szczególnie istotne, jeżeli zamierza się wykorzystać uzyskane wyniki do oceny innego systemu o podobnym typie, lecz działającego w oparciu o inny sprzęt.

Józef Czajkowski
ZOWAR
R-POLSYSTEM

Literatura

[1] Bogucki W., Staniszkis W.: Podstawowe elementy architektury systemów zarządzania bazą danych, Informatyka 1974, 10.
[2] Boot R.: User Experience with Data Base Management Systems, the Diebold Research Program, Conference Proceedings — Meeting XXXII London, November 1974.
[3] Everest G.: Data Management, Objectives, Organization and System Functions, University of Minnesota, 1974.
[4] Krinos J. D.: Interaction Statistics from a Data Base Management System, AFIPS National Computer Conference Proceedings, 1963, June.
[5] Land F.: Criteria for the evaluation and design of effective systems, International Symposium on Economics of Informatics, Proceedings Vol. september 1974.



c.d. z IV okł.

● **Organizacja maszyn cyfrowych. Laboratorium Z. 2** — PAPLIŃSKI A. STELMASIK H., ZUBEREK W. Wyd. Politechniki Warszawskiej, Warszawa 1974, s. 138, cena zł 29.— Z. 1, 2, 3. Skrypt.

Maszyna cyfrowa ODRA 1204: opis ogólny maszyny, układy podstawowe i oznaczenia, pamięć operacyjna, pamięć stała, struktura logiczna i działanie maszyny. Maszyna cyfrowa UMC-10: opis ogólny, opis schematów.
Skrypt przeznaczony jest dla studentów Wydziału Elektroniki, specjalizujących się w zakresie maszyn cyfrowych.

● **Organizacja maszyn cyfrowych. Laboratorium Z. 3 — (atlas rysunków do zeszytu 2)** — PAPLIŃSKI A., STELMASIK H., ZUBEREK W., Wyd. Politechniki Warszawskiej, Warszawa 1974, wlb, cena zł 29.— Z. 1, 2, 3. (Skrypt).
Skrypt zawiera 36 rysunków (schematów) maszyny cyfrowej ODRA 1204 i UMC-10.
Skrypt przeznaczony jest dla studentów Wydziału Elektroniki, specjalizujących się w zakresie maszyn cyfrowych.

● **Cyfrowe układy scalone TTL** — ŁAKOMY M., ZABRODZKI J. WNT, Warszawa 1974, s. 227, cena zł 24.—
Bramki TTL. Przerzutniki TTL. Układy TTL o średniej i wielkiej skali integracji. Połączenia cyfrowych układów scalonych TTL. Linie przesyłowe TTL. Zasady stosowania układów TTL.
Przykłady rozwiązań układów.
Książka przeznaczona jest dla inżynierów automatyków i elektroników projektujących układy cyfrowe.

● **Oprogramowanie maszyn cyfrowych ODRA serii 1300. EGZEKUTORY dla ODRY 1325, 1320 101.** Wyd. Wrocławskich Zakładów Elektronicznych MERA-ELWRO, Wrocław 1974, s. 84.

Cz. 1. Opis podstawowych egzekutorów maszyny cyfrowej ODRA 1325: Wprowadzenie. Funkcje egzekutora. Cz. 2. Ładowanie i start egzekutora. Ładowanie egzekutora. Start i restart egzekutora. Możliwości operatorskie kontroli pracy egzekutora. Dodatek — formularz generowania egzekutora. Cz. 3. Opisy poszczególnych egzekutorów maszyny cyfrowej ODRA 1325: Egzekutor EX 2 egzekutor EX2P.

● **Oprogramowanie maszyn cyfrowych ODRA serii 1300. FORTRAN — kompilatory.** Zeszyt 4. 1301301/4. Wyd. Wrocławskich Zakładów Elektronicznych MERA-ELWRO, Wrocław 1974, s. 33.
Struktura programu. Wejście/wyjście w programie wynikowym. Listowanie programu. Wejście kompilatora. Testowanie programu. Inne wiersze sterujące. Opis systemu. Instrukcje operatorskie. Dodatki: wykaz błędów kompilacji, wykaz błędów wykonania, kody 1900.
Materiały przeznaczone są dla programistów.

● **Oprogramowanie maszyn cyfrowych ODRA serii 1300. System NEL do obrabiarek sterowanych numerycznie.** Zeszyt 1 NEL — opis systemu język 2 CL 1303401/1. Wyd. Wrocławskich Zakładów Elektronicznych MERA-ELWRO, Wrocław 1974, s. 84.
Wprowadzenie. Program przedmiotu. Sporządzenie programu przedmiotu w języku 2 CL. Programowanie w języku 2 PL i 2 C. Przykład programu dla przedmiotu: jarzmo sprzęgła. Przetwarzanie w systemie NEL. Komunikaty o błędach. Skorowidz nazw zarezerwowanych.
Materiały przeznaczone są dla programistów i operatorów.

● **Oprogramowanie maszyn cyfrowych ODRA serii 1300. System NEL dla obrabiarek sterowanych numerycznie.** Zeszyt 4. NEL — użytkowanie systemu. 1303401/4. Wyd. Wrocławskich Zakładów Elektronicznych MERA-ELWRO Wrocław 1974, s. 38.
Wprowadzenie. Język 2 CL — wersja na taśmę magnetyczną. Język 2 CL — wersja na EDS. Język 2 PL. Język 2 C.
Materiały przeznaczone są dla programistów i operatorów.

● **Oprogramowanie maszyn cyfrowych ODRA serii 1300. Podręcznik testów maszyny cyfrowej ODRA 1304, 1305, 1325. Uzupełnienie** nr 1. 130008/1. Wyd. Wrocławskich Zakładów Elektronicznych MERA-ELWRO, Wrocław 1974, wlb.
Uzupełnienie zawiera poprawki zauważonych błędów w opisach testów, wydanych w poprzednich latach. Uzupełnienie zawiera również nowe wydanie lub wydania poprawione.
Materiały przeznaczone są dla programistów i operatorów.

● **Oprogramowanie maszyn cyfrowych ODRA serii 1300. Testy urządzeń teletransmisji danych maszyn cyfrowych ODRA 1304, 1305, 1325. Zeszyt 1. 1300005/1.** Wyd. Wrocławskich Zakładów Elektronicznych MERA-ELWRO, Wrocław 1974, wlb.

Prosty test kanału multipleksorowego. Testy: sentinela, multipleksora we/wy na dalekopis, multipleksora i unipleksora, stacji czytnik-dziurkarka, monitora ekranowego ALFA-10.
Materiały przeznaczone są dla programistów i operatorów.

● **Oprogramowanie maszyn cyfrowych ODRA serii 1300. Testy urządzeń teletransmisji danych maszyn cyfrowych ODRA 1304, 1305, 1325. Zeszyt 2. 1300005/2.** Wyd. Wrocławskich Zakładów Elektronicznych MERA-ELWRO, Wrocław 1974, wlb.
Test ogólny multipleksora i unipleksora. Test ogólny stacji czytnik-dziurkarka.
Materiały przeznaczone są dla programistów i operatorów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie DOS/JS. CEAID — program protokółujący. C13104-1.** Wyd. Wrocławskich Zakładów Elektronicznych MERA-ELWRO, Wrocław 1974, s. 19.
Wykorzystanie programu protokołowania. Formaty wydruków. Program CEAIDLST. Przykład wykorzystania programu protokołowania. Komunikaty programu protokołowania.
Materiały przeznaczone są dla programistów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie DOS/JS.** Wprowadzenie do systemu C 12001/ pervisorem C 13602-1. Wyd. Wrocławskich Zakładów Elektronicznych MERA-ELWRO, Wrocław 1974, s. 40.
Makroinstrukcje łączności z supervisorem. Makroinstrukcje CALL, SAVE, RETURN. Uzupełnienie — kody instrukcji SVC generowane w makrorozwinięciach makroinstrukcji łączności z supervisorem.
Materiały przeznaczone są dla programistów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie DOS/JS.** Wprowadzenie do systemu C 12001/ /1. Wyd. Wrocławskich Zakładów Elektronicznych MERA-ELWRO, Wrocław 1974, s. 37.
Podstawowe pojęcia. Programy sterujące. Programy serwisowe. OLTEP — testy kontrolne, pracujące pod kontrolą systemu DOS/JS. Translatory języków programowania. Generowanie systemu. Przygotowywanie i wykonywanie programów. Efektywność pracy systemu.
Materiały przeznaczone są dla programistów.

● **Jednolity System Elektronicznych Maszyn Cyfrowych. Oprogramowanie DOS/JS. Etykiety dyskowe. C** 13003-1. Wyd. Wrocławskich Zakładów Elektronicznych MERA-ELWRO, Wrocław 1974, s. 71.
Wstęp: typy etykiet, przetwarzanie etykiet. Przetwarzanie etykiet: opis zbioru, cylinder etykiet, programy użytkownika dla zbiorów SD i DA, przetwarzanie etykiet zbiorów sekwencyjnych (SD) i zbiorów losowych (DA), przetwarzanie etykiet zbiorów indeksowo-sekwencyjnych (IS). Pola etykiet.

Materiały przeznaczone są dla programistów.

● **Programowanie komputera ODRA 1204, zasady i ćwiczenia** — NIEDZIELSKA E. — red. PWE. Warszawa 1974, ss. 185.

Cz. 1. Użytkowa charakterystyka komputera ODRA 1204: Wyposażenie komputera ODRA 1204. Eksploatacja komputera. Cechy charakterystyczne języków programowania komputera ODRA 1204. Cz. 2. Programowanie komputera ODRA 1204: Przykłady i zadania w języku JAS, MOST 2 i ALGOL. Dodatki: 1. JASMOST 2. Słownik symboli stosowanych przy opisach błędów i zatrzymań.
Materiały przeznaczone są przede wszystkim dla studentów uczelni ekonomicznych oraz dla wszystkich osób interesujących się problematyką programowania komputerów.

● **Koncepcja Krajowego Systemu Informatycznego** — BOGUCKI Z. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1974, ss. 36.
Elementy składowe Krajowego Systemu Informatycznego (KSI). Przepływ informacji w KSI. Koncepcja modelu zarządzania branżą: struktura branży, etapowanie opracowań, sterowanie branżą za pomocą maszyny liczącej, precyzyjność opracowywania programów i planów, zarys modelu

zarządzania branżą. Szkic zakładowego (kombinatowego) systemu informatycznego.
Materiały przeznaczone są dla kadry kierowniczej przedsiębiorstw, zjednoczeń i resortów.

Koncepcja Krajowego Systemu Informatycznego — Bank Informacji — WOŹNICA W. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1974 ss. 22.
Założenia systemu. Istota koncepcji. Koncepcja funkcjonowania systemu. Efekty systemu. Orientacyjne nakłady realizacji systemu.
Materiały przeznaczone są dla kadry kierowniczej przedsiębiorstwa, zjednoczeń i resortów.

Informatyka we Francji — GREGORCZYK Z., MICHAŁOWSKA M., WITKOWSKI M. — oprac. Wyd. Ośrodka Badawczo-Rozwojowego Informatyki INFORNA, Warszawa 1974, ss. 96.
Charakterystyka ogólna (rola Pełnomocnika Rządu d/s Informatyki, Plan CALCUL, firma CII, IRIA itd.). Francuskie komputery. Firmy software'owe. Planowanie rozwoju informatyki. Projektowanie systemów informatycznych. Zastosowanie informatyki. Współpraca Francji z państwami Europy Zachodniej. Mała informatyka (karty selekcyjne oraz urządzenia pomocnicze do nich). Wystawy SICOB 1973. Adresy firm. Bibliografia materiałów źródłowych.
Materiały przeznaczone są dla pracowników informatyki.

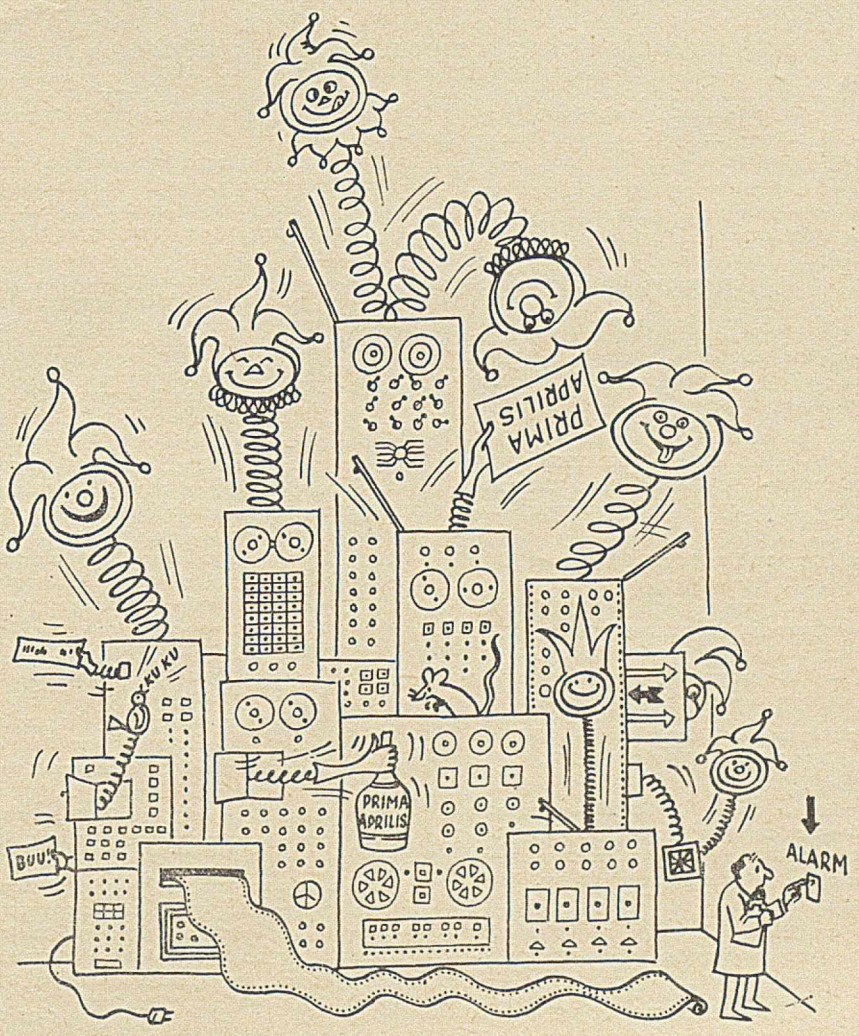
Encyklopedyczne wiadomości o maszynach cyfrowych — MIEŚCICKI J. Wyd. Instytutu Podstawowych Problemów Techniki PAN, Warszawa 1974, ss. 117. Prace JPPT 50/1974.
Dane i najogólniejsze zasady ich przetwarzania. Algorytmy, programy i języki. Język wewnętrzny maszyny cyfrowej. Organizacja typowej maszyny cyfrowej. Ogólne wiadomości o programowaniu maszyn cyfrowych.
Materiały wprowadzające przeznaczone są dla inżynierów.

Szacowanie jednorównaniowych modeli ekonometrycznych. Program w języku ALGOL-60 — JANUS S. Wyd. Szkoły Głównej Planowania i Statystyki, Warszawa 1974, ss. 76, cena zł 8 (skrypt). Przykłady systemów przetwarzania danych 1 — SZANIAWSKA M. — red.). Charakterystyka problemu, Procedury wykorzystane w programie KMNK procedura DROK, procedura OMA, testowanie procedury OMA. Program KMNK.
Skrypt przeznaczony jest dla studentów wyższych szkół ekonomicznych.

System informatyczny do zarządzania wyższą szkołą techniczną ASOS na przykładzie Politechniki Wrocławskiej — GRODZEWSKI W. M., KLONOWSKI Z. Wyd. Politechniki Wrocławskiej, Wrocław 1974, ss. 136, cena zł 16,— Biblioteka WASC.

Zarządzanie wyższą szkołą techniczną. System informatyczny dla zarządzania na poziomie kierowania operacyjnego. System informatyczny do zarządzania na poziomie kierownictwa strategicznego (SIK). Realizacja systemu.
Materiały przeznaczone są dla użytkowników systemu WASC oraz kadry kierowniczej szkół wyższych.

oprac. **J. Klamborowski**



PRZEGLĄD WYDAWNICTW

Bibliografia wydawnictw polskich z dziedziny informatyki

● **Co to jest komputer? Wprowadzenie.** Podręcznik programowany — MATHIEU G. — red. Tłum. wyd. franc. z roku 1972. PWE, Warszawa 1974, s. 161, cena zł 30.—

Wskazówki autorów dla czytelników. Co to jest przetwarzanie informacji? Kod dwójkowy — karta dziurkowana. Znak-słowo-adres. Pamięć główna i pamięci pomocnicze. Rozkazy i program. Języki i kompilacje — schematy blokowe — wejścia i wyjścia. Podsumowanie. Słowniczek terminów podstawowych.

Przystępnie napisany podręcznik, stanowiący abecadło informatyki, przeznaczony jest dla pracowników administracji i księgowości, techników i tych wszystkich, którzy zetkną się wkrótce w swojej pracy z komputerem.

● **Analiza przestrzennych konstrukcji prętowych za pomocą maszyn ICL 1900 lub ODRA 1304** — BORTKIEWICZ-STULIŃSKA C. Wyd. Politechniki Warszawskiej, Warszawa 1974, s. 142. Prace nad automatyzacją projektowania konstrukcji inżynierskich 13.

Opis działania pakietu programów. Wyniki zadania. Opis systemu. Instrukcje operacji. Przykłady obliczeń. Załączniki: 1. wzory arkuszy formularzy danych. 2. Ograniczenia wielkości zadania. 3. Wykresy operacyjne. 4. Znaczenie zwrotów angielskich przy wyprowadzaniu wyników.
Materiały przeznaczone są dla inżynierów projektantów w zakresie budownictwa lądowego oraz dla programistów.

● **Schematy blokowe algorytmów. Zbiór zadań** — SIUDAK M. Wyd. Politechniki Warszawskiej, Warszawa 1974, s. 83, cena zł 6.— (skrypt).

Etapy przygotowania obliczeń na EMC. Pojęcie algorytmu i sposoby jego przedstawiania. Zasady przyjęte przy budowie schematów blokowych. Zadania z zakresu elementarnego programowania z podziałem na cztery grupy. Część zadań podano z rozwiązaniami.

Skrypt przeznaczony jest dla studentów oraz dla osób rozpoczynających naukę programowania w dowolnym języku algorytmicznym.

● **Zbiór zadań — ALGOL 1204** — NOSOWSKI W. i inni. Wyd. Politechniki Warszawskiej, Warszawa 1974, s. 104, cena zł 7.— (skrypt).

Zadania elementarne tablicowanie funkcji. Metody numeryczne. Zadania złożone. Każde zadanie jest rozwiązane na ogół w następującym cyklu: postawienie zadania, opis metody rozwiązania, opis przyjętego algorytmu, postać danych i wyników, omówienie programu, schemat blokowy, program, dane i wyniki przykładowe.
Skrypt przeznaczony jest do użytku studentów wydziałów: Mechanicznego-Technologicznego, Mechaniki Precyzyjnej Politechniki Warszawskiej do przedmiotu Podstawy Informatyki.

● **Zbiór zadań z metod numerycznych ALGOL 1204** — ŁUBOWICZ J. i inni. Wyd. Politechniki Warszawskiej, Warszawa 1974, s. 91, cena zł 6.— (skrypt).

Zadania dotyczą metod algebry liniowej, interpolacji i aproksymacji funkcji, równań algebraicznych i przestępnych, wielomianów, całkowania numerycznego, numerycznego rozwiązywania równań różniczkowych.
Zbiór zadań przeznaczony jest przede wszystkim dla studentów wyższych uczelni i może służyć jako pomoc do zajęć z metod numerycznych oraz programowania komputerów.

● **Organizacja maszyn cyfrowych. Laboratorium Z. 1.** — PAPLIŃSKI A., STELMASIK H., ZUBEREK W. Wyd. Politechniki Warszawskiej, Warszawa 1974, s. 161, cena zł 29. — Z. 1, 2, 3, (skrypt).

Arytmometr maszyn cyfrowych: zapisy liczbowe, algorytmy mnożenia i dzielenia, przykładowy arytmometr równoległy, arytmetyka zmienno-przecinkowa, zapis redundacyjny i residualny, przykłady. Organizacja współpracy z urządzeniami peryferyjnymi: kanały nieautonomiczne, autonomiczne i standardowe (Systemu Jednolitego oraz serii ODRA 1300).
Skrypt przeznaczony jest dla studentów Wydziału Elektroniki, specjalizujących się w zakresie maszyn cyfrowych. Stanowi materiał podstawowy dla zajęć laboratoryjnych i projektowych przedmiotów „Organizacja Maszyn Cyfrowych" oraz „Organizacja Maszyn i Systemów Cyfrowych".

ciąg dalszy str. 48, III okł.