

P. 2900/73

# MERA

AUTOMATYKA PRZEMYSŁOWA

APARATURA POMIAROWA

INFORMATYKA



# BIULETYN

12 (142)  
Rok XII 1973



## KOLEGIUM REDAKCYJNE

Redaktor Naczelny: mgr Roman Sprawski  
Sekretarz Redakcji: mgr Zofia Bieguszevska-Kochan  
Redaktorzy działowi: mgr Bolesław Drożak  
mgr inż. Andrzej Wyrzykowski  
Jan Grzędzielski  
Członkowie: mgr inż. Ryszard Jackowicz  
mgr inż. Henryk Chyrek  
mgr Czesław Pawlak  
mgr inż. Ludomir Krzystalik  
inż. Ludomir Kowalski

## WARUNKI PRENUMERATY

Cena prenumeraty rocznej - 516,- zł

Instytucje państwowe i społeczne mogą zamawiać prenumeratę wyłącznie za pośrednictwem Oddziałów i Delegatur CKPiW "RUCH". Prenumeraty dla czytelników indywidualnych przyjmują urzędy pocztowe oraz listonosze. Można również dokonać wpłat na konto PKO nr 1-6-100020 CKPiW "RUCH", Warszawa, ul. Wronia 23

ZJEDNOCZENIE PRZEMYSŁU AUTOMATYKI  
I APARATURY POMIAROWEJ „MERA”



# BIULETYN „MERA”

AUTOMATYKA PRZEMYSŁOWA  
APARATURA POMIAROWA  
I N F O R M A T Y K A

WARSZAWA, GRUDZIEŃ 1973



Spis treści

- M. J. Greniewski	Wytwarzanie oprogramowania metodami przemysłowymi .....	3
- A. M. Wiśniewski	Systemy operacyjne JS EMC .....	12
- A. M. Wiśniewski S. J. Podlewski	Komputer biurowy MERA 302 - architektura i zasady działania	18
- H. Orłowski	Oprogramowanie komputerów w układach automatyki i pomiarów .....	26
Spis artykułów opublikowanych w Biuletynie "Mera" w roku 1973 .....		31

## WYTWARZANIE OPROGRAMOWANIA METODAMI PRZEMYSŁOWYMI

### 1. Wprowadzenie do problematyki produkcji oprogramowania

Prawdopodobnie pierwszymi pytaniami, jakie nasuną się Czytelnikowi, będą pytania:  
- Czy tytuł nie jest jakimś nieporozumieniem? Czy istnieje coś takiego, jak "produkt programowy"? Programowanie to chyba usługa? W niniejszym artykule autor będzie starał się przekonać Czytelnika, że można mówić o produkcji oprogramowania oraz że oprogramowanie tak wytworzone ma cechy produktu przemysłowego.

Rozwinięcie zastosowań komputerów wiąże się ściśle z koniecznością stworzenia odpowiednio elastycznego i bogatego oprogramowania. Stworzenie tego rodzaju oprogramowania wymaga przejścia na przemysłowe metody wytwarzania oprogramowania.

Przemysłowe metody wytwarzania oprogramowania opierają się o:

- technologię wytwarzania oprogramowania,
- technologiczną konstrukcję oprogramowania,
- oprzyrządowanie programowe,
- sprzęt,
- programistów profesjonalnych.

Technologia wytwarzania oprogramowania dotyczy oprogramowania przeznaczonego do wielokrotnego wykorzystania przez różnych użytkowników. Ma ona na celu zapewnienie:

- uniwersalności oprogramowania,
- odpowiedniej jakości i efektywności oprogramowania,
- możliwości wprowadzenia modyfikacji i dalszej rozbudowy,
- zwiększenia wydajności pracy programistów profesjonalnych,

- szczegółowego planowania prac i na tej drodze - zmniejszenie niepewności związanej z terminami i kosztami wytworzenia oprogramowania,
- standaryzacji form dokumentacji.

Należy podkreślić, że znajomość technologii wytwarzania oprogramowania w Polsce jest niemal zerowa. Pewne doświadczenia w tym zakresie wynikające z ponad rocznej działalności Zakładu Doświadczalnego Oprogramowania przy Instytucie Maszyn Matematycznych ma Zjednoczenie Przemysłu Automatyki i Aparatury Pomiarowej "MERA".

Technologiczne konstrukcje oprogramowania są to konstrukcje, składające się z typowych elementów łączonych w standardowy sposób i charakteryzujące się tym, że dają się podzielić na komponenty /zespoły główne/, które mogą być niezależnie projektowane, uruchamiane, integrowane i testowane.

Programowe oprzyrządowanie procesu wytwarzania oprogramowania składa się między innymi z następujących języków i systemów programowania:

- języków i kompilatorów przenaszalnego oprogramowania,
- systemu bazy danych i pakietów manipulacyjno-dokumentacyjnych typu PPL /Program Production Library/ - niezbędnych do stosowania metod programowania strukturalnego,
- systemów modelowania struktury produktu,
- biblioteki typowych modułów działającej pod nadzorem systemu manipulacyjno-dokumentacyjnego,
- systemów testujących współdziałających z systemem manipulacyjno-dokumentacyjnym,



- systemów wspomaganego projektowania i budowania systemów zastosowaniowych /tzw. Meta-Systemy/.

Należy podkreślić, że obecny stan programowego oprzyrządowania procesu wytwarzania oprogramowania jest ciągle jeszcze bliski zera. Potrzeba jeszcze kilku lat intensywnych prac, aby stworzyć odpowiednio bogate oprzyrządowanie programowe.

Niezbędny sprzęt do wytwarzania oprogramowania, to przede wszystkim odpowiednie komputery wyposażone w terminale programowane, pamięci masowe wielkiej pojemności, urządzenia do automatyzacji składu drukarskiego, graficznego zobrazowania itd. Ponadto urządzenia powielające, drukarskie, introligatorskie itp. Ocenia się, że dla zorganizowania procesu wytwarzania oprogramowania metodami przemysłowymi, dla jednego programisty zatrudnionego przy wytwarzaniu oprogramowania niezbędne jest zainstalowanie sprzętu wartości rzędu 2 mln zł. Tak więc, jednostka produkcji oprogramowania zatrudniająca 250 programistów winna posiadać sprzęt o wartości rzędu 450 - 500 mln zł. Należy podkreślić, że sprzęt taki powinien ulegać odnowie w okresach 4-5 lat.

Sprzęt i oprzyrządowanie programowe tworzą łącznie instalację zwaną Linią Produkcji Oprogramowania.

Przemysłowe metody wytwarzania oprogramowania wymagają wykształcenia nowego typu programisty, zwanego dalej programistą profesjonalnym. W odróżnieniu od dominującego obecnie typu programisty amatora, programista profesjonalny musi umieć: pracować zespołowo przy użyciu Linii Produkcji Oprogramowania i programować metodami przenaszalnymi z przestrzeganiem zasad określonych technologią wytwarzania oprogramowania. Programista profesjonalny musi także umieć biegle programować w kilku różnych językach programowania oraz posługiwać się językiem dokumentacyjnym.

Przedstawione w niniejszym artykule problemy organizacji produkcji oprogramowania dotyczą oprogramowania przeznaczonego do wielokrotnego wykorzystania.

## 2. Produkt programowy

W dalszym ciągu niniejszego artykułu każdy system programowania lub pakiet programów wytwarzany metodami przemysłowymi będziemy nazywali produktem programowym. Produkt składa się z następujących części:

- wymagań technicznych na produkt,
- kodu źródłowego produktu / x /,
- kodu wynikowego produktu w postaci dystrybucyjnej,

- kodu źródłowego specjalnego oprzyrządowania programowego /nie zawsze występuje/ /x /,
- kodu wynikowego specjalnego oprzyrządowania programowego /nie zawsze występuje/,
- specyfikacji produktu,
- krótkiego opisu wprowadzającego,
- przewodnika użytkownika,
- przewodnika konserwacji / x /,
- instrukcji dla programisty systemowego,
- instrukcji dla operatora,
- instrukcji przygotowania danych,
- dokumentacji konstrukcyjnej produktu / x /,
- programu szkolenia użytkownika,
- pomocy audiowizualnych do szkolenia,
- przykładów do demonstracji,
- zadań kontrolnych.

Przedmiotem sprzedaży mogą być wszystkie części produktu, w przypadku gdy producent nie prowadzi konserwacji produktu we własnym zakresie /x/ lub - części nie oznaczone gwiazdką - w przypadku gdy producent prowadzi konserwację. Udział kosztów w poszczególnych etapach kształtuje się dla systemów operacyjnych i systemów przetwarzania danych jak w tabelicy 1.

T a b l i c a 1

Nr etapu	Nazwa etapu wytwarzania produktu	Procentowy udział w kosztach
1	Rozpoznanie i opracowanie wymagań	10%
2	Projektowanie koncepcyjne	13%
3	Test alfa	6%
4	Projektowanie szczegółowe, kodowanie, uruchamianie, integracja i dokumentowanie	20%
5	Test beta	26%
6	Przygotowanie do rozpowszechniania	5%
7	Konserwacja produktu	20%
R a z e m		100%

Przez cykl wytwarzania produktu rozumie się czas realizacji etapów od 1 do 6 łącznie.

Przykładowo, wg dotychczasowych doświadczeń cykl wytwarzania translatorów i programów sterujących systemem operacyjnego wynosi średnio 24 miesiące, przy czym w zależności od stopnia złożoności zadania może się wahać od 10 do 44 miesięcy; zaś cykl wytwarzania systemów zastosowaniowych i pakietów programów wynosi średnio 18 miesięcy, przy czym w zależności od



stopnia złożoności zadania może się wahać od 8 do 27 miesięcy.

Każdy produkt składa się z pewnej ilości komponentów /zespołów głównych/, które ze względu na swoją strukturę logiczną oraz ilość i rodzaj urządzeń peryferyjnych, mogą być zaliczone do jednej z pięciu klas złożoności. W tabelicy 2 przedstawione są przykłady komponentów zaliczonych do poszczególnych klas.

Tablica 2

Lp. Typ komponentu	Przykład
1. Prosty	Program drukowania sprawozdań o stanach i obrotach magazynowych
2. Średnio złożony	Program aktualizacji zbioru danych wieloma typami rekordów
3. Złożony	Translator
4. Bardzo złożony	Supervizor
5. Specjalny	Program obsługi terminali w reżimie interakcyjnym

Cena produktu jest wyznaczona przez sumowanie cen komponentów składających się na dany produkt. Z kolei cena komponentu jest określona w oparciu o rozmiar komponentu sprowadzony np. do ilości rozkazów języka typu Assembler przy użyciu tablic przeliczeniowych i cenę jednostkową jednego rozkazu dla komponentu danej klasy złożoności. Tablica 3 zawiera przykładowo współczynniki zamiany - rozmiarów komponentów programów pisanych w różnych językach.

Tablica 3

Współczynniki zamiany rozmiarów komponentów		Rozmiar komponentu pisanego			
		PLAN	ASSEMBLER	FORT-RAN	COBOL
Ocena rozmiarów	PLAN	1,0	0,7	0,25	0,3
	ASSEMBLER	1,4	1,0	0,3	0,4
	FORT-RAN	4,0	3,33	1,0	-
	COBOL	3,0	2,5	-	1,0

U w a g a: w przypadku, gdy dla wytworzenia produktu konieczne jest opracowanie specjalnego oprzyrządowania, cena zbytu jest obliczana łącznie, na podstawie stopnia złożoności i rozmiarów komponentów produktu i specjalnego oprzyrządowania.

### 3. Proces wytwarzania produktu programowego

Zgodnie z tym co przedstawia tablica 1, proces wytwarzania produktu programowego składa się z siedmiu etapów, które zostaną poniżej kolejno omówione.

3.1. Rozpoznanie i opracowanie wymagań dotyczących produktu. Realizując ten etap, należy znaleźć odpowiedź na następujące pytania:

- Co mogłoby być produkowane?
- Do jakich produktów ma to być podobne?
- Jacy są potencjalni użytkownicy przyszłego produktu?
- Jak ocenia się wstępnie koszt produktu?

W wyniku realizacji tego etapu powstają wymagania techniczne na produkt.

Po zakończeniu etapu należy podjąć decyzję będącą odpowiedzią na pytanie: Czy można to wyprodukować? W przypadku odpowiedzi pozytywnej, rozpoczyna się realizowanie drugiego etapu.

3.2. Projektowanie koncepcyjne produktu. Realizując ten etap, należy znaleźć odpowiedź na następujące pytania:

- Jak produkt powinien być zbudowany, jaki zawiera nukleus i jakie kolejne warstwy nadbudowane?
- Jakie istniejące komponenty lub moduły biblioteczne mogą być wykorzystane?
- W jakich językach programowania produkt powinien być pisany?
- Jak produkt powinien być testowany?
- Jakie są rozmiary i skala trudności poszczególnych komponentów produktu?

W wyniku realizacji tego etapu powstają pierwsze wersje: specyfikacji produktu, dokumentacji konstrukcyjnej produktu i wymagań na specjalne oprzyrządowanie programowe produktu.

3.3. Test alfa. Realizując ten etap, należy znaleźć odpowiedź na następujące pytania:

- Czy projektowany produkt będzie spełniał wymagania techniczne?
- Czy zaprojektowana struktura produktu jest właściwa ze względu na realizację i konserwację?
- Czy proponowana metoda testowania produktu jest adekwatna do jego struktury?
- Jak ocenia się koszty produktu?

W toku realizacji tego etapu wprowadza się szereg zmian do wyników poprzedniego etapu. W wyniku powstają kolejne wersje: kosztorysu produktu, specyfikacji produktu,

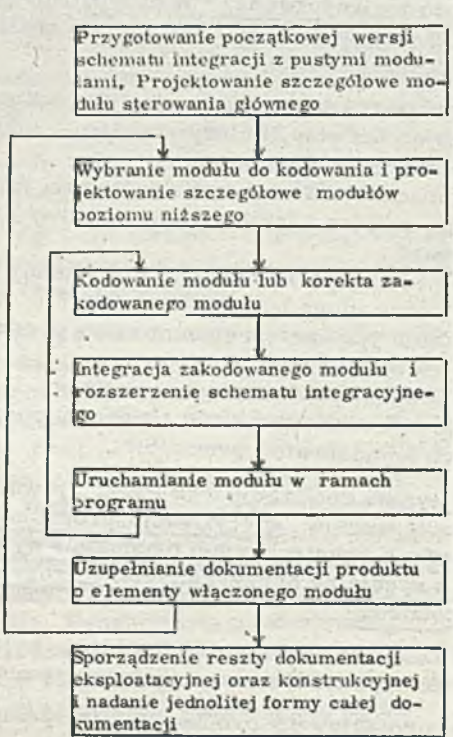


dokumentacji konstrukcyjnej produktu i wymagań na specjalne oprzyrządowanie programowe produktu. Ponadto powstają wymagania na zadania kontrolne.

Po zakończeniu etapu należy podjąć decyzję będącą odpowiedzią na pytanie: Czy należy projekt realizować? W przypadku odpowiedzi pozytywnej, rozpoczyna się realizacja czwartego etapu.

3.4. Projektowanie szczegółowe, kodowanie, uruchamianie, integracja i dokumentowanie. Etap ten jest realizowany według schematu 1. W wyniku wykonania tego etapu powstają następujące części produktu:

- kod źródłowy produktu,
- kod wynikowy produktu w postaci dystrybucyjnej,
- kod źródłowy specjalnego oprzyrządowania programowego /nie zawsze występuje/,



Schemat 1 Praca przy stosowaniu metody programowania strukturalnego CPT /Chief Programmer Team/

- kod wynikowy specjalnego oprzyrządowania programowego /nie zawsze występuje/,
- przewodnik użytkownika produktu,
- przewodnik konserwacji produktu,
- instrukcja dla programisty systemowego,
- instrukcja dla operatora,
- instrukcja przygotowania danych,
- dokumentacja konstrukcji produktu

3.5. Test beta. Realizując ten etap, należy znaleźć odpowiedź na następujące pytania:

- Czy produkt spełnia wymagania techniczne?

- Czy dokumentacja użytkowa wystarcza dla posługiwania się produktem?
- Czy dokumentacja konstrukcyjna wystarcza dla konserwacji produktu?
- Ile wyniosły dotychczasowe koszty wytworzenia produktu?

W toku realizacji tego etapu opracowuje się, według wymagań, zadania kontrolne, sprawdza się poprawność i przejrzystość dokumentacji, wprowadza się konieczne zmiany, uzupełnienia itp.

Po zakończeniu etapu należy odpowiedzieć na pytanie: Jak i kiedy produkt ma być rozpowszechniany?

3.6. Przygotowanie do rozpowszechniania. Realizując ten etap, należy znaleźć odpowiedź na następujące pytania:

- Jak demonstrować produkt?
- Jak szkolić użytkowników?
- Kto i gdzie będzie szkolił użytkowników?
- Kto będzie konserwował produkt?

W wyniku realizacji etapu powstają następujące części produktu:

- krótki opis wprowadzający,
- przykłady do pokazu,
- program szkolenia użytkownika
- pomoce audiowizualne do szkolenia

Ponadto, w ramach tego etapu następuje wydanie dokumentacji w odpowiednich nakładach.

3.7. Konserwacja produktu. Jest to etap realizowany przez okres kilku lub więcej lat. W toku realizacji trzeba możliwie szybko reagować na sygnały pochodzące od użytkowników i odpowiadać na jedno z dwu pytań: Jak należy zlokalizować i usunąć wykryty błąd? Jak zwiększyć efektywność produktu?

#### 4. Baza elementowa

Elementem konstrukcyjnym programu jest tzw. makros. Makrosy wchodzące w skład bibliotek i przeznaczone do wielokrotnego wykorzystania nazywamy modułami. Makrosy lub moduły pisane w języku wysokiego rzędu dzielą się na:

- opisujące strukturę danych,
- funkcyjne.

W dalszych rozważaniach ograniczymy się do przedstawienia jedynie tych ostatnich, nazywając je w skrócie elementami funkcyjnymi. Każdy element /makros lub moduł/ funkcyjny posiada jedno wejście i jedno wyjście. Przez wejście i wyjście rozumiemy trwałe przekazanie sterowania. Każdy element funkcyjny, ma swoją nazwę jednoznaczną w obrębie komponentu. Rozmiar elementu funkcyjnego nie przekracza 50 - 100 linii kodowych w języku wysokiego rzędu. Stero-



wanie do elementu funkcyjnego może być przekazane:

a/ Z ustawieniem śladu, według którego nastąpi powrót do elementu wołającego po wykonaniu czynności realizowanych przez dany element funkcyjny.

b/ Z ustawieniem śladu, według którego nastąpi powrót do elementu wołającego dopiero po wykonaniu czynności realizowanych przez wskazany element funkcyjny, z przekazaniem sterowania do kolejnego elementu funkcyjnego umieszczonego za danym elementem po jego realizacji.

c/ Bez ustawiania śladu, z przekazaniem sterowania do kolejnego elementu funkcyjnego umieszczonego za danym elementem po jego realizacji.

d/ Bez ustawienia śladu, z przekazaniem sterowania według śladu wcześniej ustawionego do realizacji danego elementu.

Elementy funkcyjne ustawione w programie w kolejnych obszarach pamięci mogą przekazać sobie wzajemnie sterowanie według zasady b/, c/ lub d/ - tworzą tzw. sekwencję elementów funkcyjnych. Dovolny element funkcyjny może zawierać instrukcję /jedną lub więcej/ wołania dowolnego elementu funkcyjnego lub sekwencji elementów funkcyjnych. Przez wołanie rozumieamy chwilowe przekazanie sterowania do danego elementu lub sekwencji elementów z powrotem przez ślad.

Każdy element funkcyjny jest dokumentowany według zasady przedstawionej na schemacie 2

NAZWA ELENENTU FUNKCYJNEGO
LISTA nazw elementów funkcyjnych współpracujących bezpośrednio z danym elementem
OBSZAR DANYCH WEJŚCIOWYCH - lista nazw danych argumentów dla danego elementu funkcyjnego
OBSZAR DANYCH WYJŚCIOWYCH - lista nazw danych - wyników dla danego elementu funkcyjnego
OBSZAR DANYCH ROBOCZYCH - lista nazw danych efemerydalnych
OBSZAR DANYCH PAMIĘTANYCH - lista nazw danych przechowywanych między kolejnymi realizacjami elementu
AKCJE - lista akcji powodujących przekazanie sterowania do elementu
FUNKCJE - opis czynności realizowanych przez dany element

Schemat 2

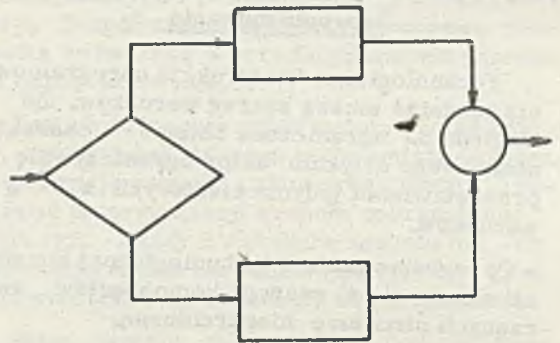
U w a g a: element funkcyjny nie może zmieniać się w wyniku kolejnych realizacji.

Każdy element funkcyjny może mieć jedną z pięciu struktur logicznych:

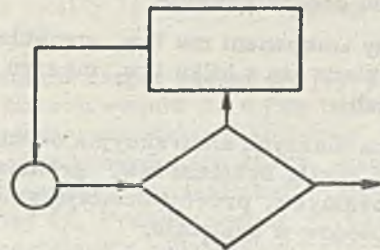
- Sekwencja operacji /schemat 3/,
- IF - THEN-ELSE /schemat 4/,
- DO - While /schemat 5/,
- DO - UNTIL /schemat 6/,
- CASE /schemat 7/



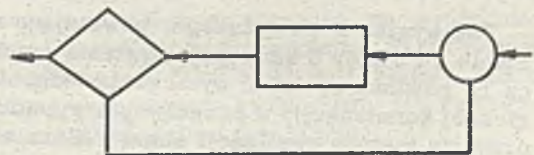
Schemat 3



Schemat 4

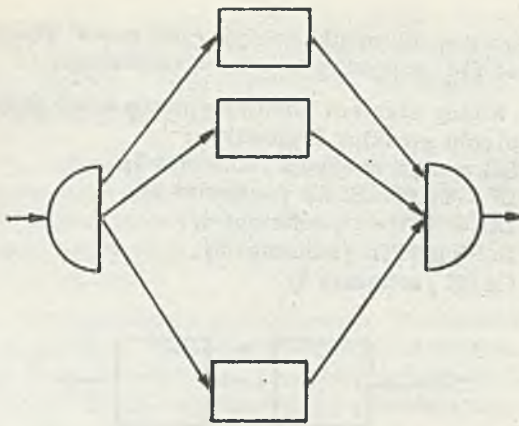


Schemat 5



Schemat 6





Schemat 7

## 5. Technologiczne konstrukcje oprogramowania

Technologiczne konstrukcje oprogramowania spełniać muszą szereg warunków. Ze względu na ograniczone ramy i charakter niniejszego artykułu autor ograniczy się do przedstawienia jedynie niektórych z tych warunków.

- Oprogramowanie o technologicznej konstrukcji składa się z szeregu komponentów tworzących strukturę hierarchiczną.
- Komponent każdej warstwy może kontaktować się bezpośrednio co najwyżej z dwoma komponentami nadrzędnym i podrzędnym /jeśli ten ostatni istnieje/. Komponent najniższej warstwy nazywamy nukleusem danego systemu programowania.
- Każdy komponent ma tzw. strukturę ziarnistą i składa się z kilku tzw. maszyn abstrakcyjnych.
- Każda maszyna abstrakcyjna oddzielona jest od reszty systemu tzw. grodziami błędoszczelnymi, przeciwdziałającym propagacji błędów w systemie.
- Każda maszyna abstrakcyjna realizuje jedno z typowych działań na strukturach danych
- Każda maszyna abstrakcyjna składa się z elementów funkcyjnych /omawianych w poprzednim rozdziale/.
- Podział oprogramowania na komponenty jest optymalny ze względu na założoną funkcję celu.

Jak wynika z powyższego, powstająca na etapie 2 i etapie 3 koncepcja systemu /tablica 1/ powinna spełniać wymagania technologicznej konstrukcji; w przeciwnym wypadku przyjęta metoda realizacji etapu 4 niezależnego budowania poszczególnych komponentów nie da się zastosować.

## 6. Standardy produkcji oprogramowania

Wprowadzenie standardów produkcji oprogramowania jest niezbędne dla zapewnienia efektywnego działania przedsiębiorstwa produkcji oprogramowania. Cele standardów są następujące:

- Zabezpieczenie użytkownika przed wytwarzaniem złej jakości produktu programowego.
- Zabezpieczenie programisty profesjonalnego przed postawieniem mu niejednoznacznie sformułowanego zadania.
- Jednoczesne określenie podziału zadań między pracownikami zatrudnionymi bezpośrednio lub pośrednio przy produkcji, wydawnictwach, szkoleniu i obsłudze użytkowników.
- Umożliwienie planowania poszczególnych etapów wytwarzania oprogramowania i kontrolę przebiegu realizacji.
- Zmniejszenie niekorzystnych dla przedsiębiorstwa produkcji oprogramowania efektów płynności kadr, przez zapewnienie możliwości przejęcia i kontynuowania pracy przez innych pracowników.
- Umożliwienie szybkiej modyfikacji istniejących produktów i wykorzystanie w nowych produktach komponentów lub modułów produktów wcześniej wytwarzanych.
- Stworzenie podstaw elastycznego organizowania zespołów zadaniowych.
- Ułatwienie szkolenia nowej kadry przedsiębiorstwa produkcji oprogramowania.

Standardy produkcji oprogramowania dzielą się na pięć grup:

- a/ Standardy etapów i czynności wytwarzania produktu programowego,
- b/ Standardy kompletacji i zawartości poszczególnych części produktu programowego.
- c/ Standardy formułowania warunków technicznych na produkt programowy.
- d/ Standardy konstrukcyjne produktu programowego.
- e/ Standardy oprzyrządowania programowego.

Poniżej omówione zostaną poszczególne grupy standardów.

6.1. Standardy etapów i czynności wytwarzania produktu programowego. Przykładem standardów tej grupy są przedstawione w rozdziale 3 etapy wytwarzania produktu programowego. Oczywiście jest, że szczegółowość standardów jest znacznie większa od zakresu przedstawionego w rozdziale 3.



Do tej grupy standardów zaliczamy również schematy metodyczne pracy, takie jak schemat metody programowania strukturalnego CPT /rozdział 3/. Do tej samej grupy standardów zaliczamy tablice wydajności programisty profesjonalnego, uzupełnione kryteriami zaliczenia komponentu do danej grupy oraz tablice przeliczania rozmiarów komponentów pisanych w różnych językach programowania i tablice przeliczania wydajności programistów profesjonalnych różnych kategorii.

6.2. Standardy komplekacji i zawartości poszczególnych części produktu programowego. Przykładem standardu na skład produktu programowego jest lista komplekcyjna produktu programowego przedstawiona w rozdziale 2. Dla każdej części produktu programowego istnieje standard zawartości, określający co ma zawierać określona część produktu i w jakim układzie. Szczególnym przypadkiem standardu zawartości jest standard na komentarze kodu źródłowego produktu i kodu źródłowego oprzyrządowania specjalnego.

6.3. Standardy formułowania warunków technicznych na produkt programowy. Standardy te w jakimś sensie są rozwinięciem fragmentu standardów omawianych w pkt 6.2., a mianowicie zawartości wymagań technicznych na produkt programowy. Mają one na celu scharakteryzowanie ilościowo-jakościowe produktu programowego.

6.4. Standardy konstrukcyjne produktu programowego. Standardy konstrukcyjne mają na celu określenie struktury produktu programowego. Do tej grupy standardów zaliczamy:

- Zasadę konstruowania produktu w formie "nucleus" i nadbudowanych nad nim kolejnych warstw.
- Zasadę "ziarnistej" budowy nucleusa i warstw.
- Zasadę realizowania nucleusa i kolejnych warstw produktu jako komponentów produktu.
- Zasadę modularnego budowania komponentów zgodnie z metodą programowania strukturalnego CPT oraz bazą elementów funkcjonalnych.
- Zasady budowy bibliotek modułów i komponentów oraz ich użytkowania.
- Standardy na struktury danych.

Ponadto, do grupy tej zaliczamy standardy na łączenie modułów i komponentów produktu programowego.

6.5. Standardy oprzyrządowania programowego. Do tej grupy standardów zaliczamy standardy na reprezentację języków dokumentacyjnych i języków programowania oraz ich wzorcowych implementacji np. COBOL, ANS, FORTRAN, APL, BASIC, DETAB, FOR-TAB, DBMS i inne/, a w szczególności stan-

dardy na makrogeneratory i ich wzorcowe implementacje /np. STAGE II, PML i inne/. Do tej grupy również zaliczamy standardy na języki manipulacji bibliotekami typu PPL i inne elementy programowego oprzyrządowania Linii Produkcji Oprogramowania.

Standardy są podstawą opracowywania metodyk, takich jak:

- Oceny rozmiarów i złożoności planowanych do wytworzenia produktów.
- Planowania wytworzenia produktu programowego.
- Realizacji poszczególnych etapów wytwarzania produktu programowego.
- Badań jakościowych produktu programowego.

### 7. Zespół zadaniowy

Zespół zadaniowy składa się z programistów profesjonalnych i /ewentualnie/ pracownika obsługi. Zespół zadaniowy jest podstawową jednostką wytwórczą w przedsiębiorstwie produkcji oprogramowania.

Zespołem kieruje programista profesjonalny, zwany dalej głównym programistą zespołu. Organizację zespołu zadaniowego można porównać do organizacji zespołu operacyjnego chirurgii. Każdy z członków zespołu ma do wykonania ściśle określone funkcje i jest odpowiedzialny za określony aspekt produktu.

Skład zespołu zadaniowego i funkcje poszczególnych członków są różne na różnych etapach powstawania produktu. Dlatego zespół zadaniowy może liczyć od 3 do 12 osób. Na etapach 4 i 5 /wg tablicy 1/ zespół zadaniowy liczy od 4 do 12 osób. Natomiast na etapach 2 i 3 /wg tablicy 1/ zespół zadaniowy liczy od 3 do 6 osób.

W zasadzie zespół zadaniowy jest powoływany do realizacji etapów 2 i 4 /wg tablicy 1/ oraz współuczestniczenia z komórką kontroli jakości oraz komórką struktur i standardów oprogramowania przy realizacji etapów 3 i 5 /wg tablicy 1/. Ponadto zespół zadaniowy współuczestniczy z komórką wdrażania produktów i komórką wydawniczą przy realizacji etapu 6 /wg tablicy 1/.

W uzasadnionych przypadkach, zespół zadaniowy może zostać powołany na etapie 1 /wg tablicy 1/ i współuczestniczyć z komórką badania rynku i komórką informacji techniczno-ekonomicznej.

Zespół zadaniowy jest powoływany dla wykonania określonego zlecenia, na wyprodukowanie określonego komponentu. Skład zespołu zadaniowego w toku realizacji etapu może ulegać okresowemu zwiększeniu lub zmniejszeniu w zależności od frontu prac.

Przedstawimy przykładowo zadania poszczególnych członków zespołu zadaniowego



przy realizacji etapu 4 /wg tablicy 1/. Etap 4 jest realizowany zgodnie ze schematem 1.

W skład minimalnego zespołu zadaniowego wchodzi:

- główny programista zespołu,
- programista asystent,
- programista analityk,
- programista operator PRL /Program Production library/.

W miarę potrzeb zespołu, zespół zadaniowy zostaje rozszerzony o dalszych programistów i ewentualnie programistów analityków.

Do zadań głównego programisty przy realizacji etapu 4 oprócz kierowania całością prac zespołu zadaniowego, należy projektowanie szczegółowej struktury logicznej produktu oraz schematu integracji komponentu.

Do zadań programisty asystenta należy kodowanie prostszych modułów i uzupełnienie dokumentacji produktu.

Do zadań programisty analityka należy projektowanie szczegółowe całej sygnalizacji komponentu i kodowanie modułów realizujących tę sygnalizację.

Do zadań programisty operatora PPL należy operowanie biblioteką /PPL/ przy użyciu terminala dla potrzeb zespołu zadaniowego, w celu integracji i dokumentowania modułów produktu.

Do zadań pozostałych programistów należy kodowanie i uzupełnianie bardziej złożonych modułów lub przygotowywanie danych testujących dla uruchomienia modułów.

## 8. Linia Produkcji Oprogramowania

Dotychczasowe rozważania pozwalają sprecyzować wymagania na linię Produkcji Oprogramowania /w skrócie LPO/. Na przykładzie poprzedniego rozdziału pokazano, w jaki sposób użytkuje LPO zespół zadaniowy, realizując etap 4 /tablica 1/. Z rozważań rozdziału 5 wynika, jakie wymagania stawia przed LPO realizacja etapu 2 i etapu 3.

W świetle dotychczasowych rozważań widać jasno, że LPO musi być:

- Węzłem informacyjnym zapewniającym koordynację całej działalności twórczej przedsiębiorstwa produkcji oprogramowania.
- Urządzeniem umożliwiającym interakcyjne manipulowanie kodami elementów funkcyjnych i dokumentacją konstrukcyjną komponentów i produktów w zakresie niezbędnym dla stosowania np. metody CPT.
- Urządzeniem umożliwiającym badanie przez modelowanie optymalności konstrukcji produktu.
- Urządzeniem umożliwiającym interakcyjne kompilowanie i generowanie w oparciu o me-

tody przenaszalnego oprogramowania.

- Urządzeniem umożliwiającym symulowanie szerokiej klasy komputerów /tzw. zasada First Software Computer/.
- Urządzeniem umożliwiającym podłączenie metodą Chanel to Chanel Adapter dowolnego komputera, dla którego wytwarzane jest oprogramowanie do LPO, dla przesyłania testowanych fragmentów produktu z LPO do komputera i odwrotnie.
- Urządzeniem automatyzującym skład drukarski dokumentacji produktu.

Jak wynika z powyższego, LPO powinna składać się z następującego sprzętu:

- Szybkiego procesora wyposażonego w dużą pamięć operacyjną i stosunkowo liczne kanały.
- Pamięci dyskowej o pojemności rzędu 500 Mbajtów.
- Około 40 terminali programowanych wyposażonych w małe pamięci dyskowe, urządzenia drukujące, specjalizowane klawiatury itp.
- Około 10 terminali monitorów - ekranowych.
- Adaptery kanałów umożliwiające podłączenia innych procesorów do LPO.
- Urządzenia do automatyzacji składu drukarskiego.
- Klasycznych urządzeń peryferyjnych.

Terminale powinny być podzielone na 4 grupy - zgodnie ze sposobem ich użytkowania:

- terminale programowane wykorzystywane przez zespoły zadaniowe programistów produkujących oprogramowanie;
- terminale do prac redakcyjnych nad dokumentacją prowadzone przy współudziale komórki wydawniczej;
- terminale programowe wykorzystywane dla przygotowania, wdrożeń i szkolenia użytkowników przez komórkę wdrażania i komórkę szkolenia;
- terminale wykorzystywane dla potrzeb zarządzania produkcją.

Na Linię Produkcji Oprogramowania składać się będzie następujące oprzyrządowanie programowe:

- Specjalizowany system operacyjny dla LPO zapewniający efektywną obsługę terminali, współpracę z bibliotekami dyskowymi i dwustronne przekazywanie zadań i przyjmowanie wyników z podłączonych przez adaptery kanał-kanał pozostałych komputerów;
- System obsługi bazy danych dostosowany do współpracy z PPL dla efektywnego prowadzenia bibliotek dokumentacji i programów;
- Pakiety manipulowania elementami bibliotek /PPL/ realizujące między innymi następujące funkcje: wyszukiwanie, włączanie według zadanych kontekstów, rozrzucanie i łączenie tekstów z bibliotek;



- Zestaw niezależnych kompilatorów dla języków programowania systemowego i dokumentowania;
- Pakiet programów dla sprawdzania poprawności programów napisanych w konwencji programowania strukturalnego /metoda CPT/;
- Pakiety programów dla tworzenia dokumentacji eksploatacyjnej i konstrukcyjnej oraz jej aktualizowania;
- Systemy wspomaganego budowania systemów zastosowaniowych;
- Pakiet programów modelowania działania projektowanego produktu.

#### 9. Uwagi końcowe

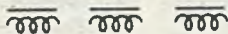
Jak wynika z dotychczasowych rozważań zorganizowanie wytwarzania oprogramowania metodami przemysłowymi jest przedsięwzięciem bardzo złożonym. Prowadzone obecnie prace nad stworzeniem w Zjednoczeniu MERA pierwszej Linii Produkcji Oprogramowania zostaną zakończone w roku 1976. Eksperymentalne wykorzystywanie LPO rozpocznie się na początku roku 1975.

W wyniku dotychczas prowadzonych prac zostały ukształtowane poglądy na problematy-

kę produkcji oprogramowania oraz przygotowania została dostatecznie liczna kadra programistów profesjonalnych, niezbędna dla podjęcia produkcji oprogramowania metodami przemysłowymi.

#### Literatura

- [1] A. Detailed Review of Chief Programmer Team Operations, IBM FSC 72 - 5177, Federal System Center 18100 Frederick Pike Gaithersburg, Maryland 20760.
- [2] B. W. Boehm, Software and its Impact A Quantitative Assessment Datamation. May 1973, pp 48-59.
- [3] J. Duńda i A. Wiśniewski: Problematyka Projektowania Systemów Oprogramowania - Potrzeby i Dotychczasowe Rezultaty w Projektowanie Maszyn i Systemów Cyfrowych, PWN, Warszawa 1972.
- [4] W. M. Turski: Projektowanie Oprogramowania Systemów Liczących w Projektowanie Maszyn i Systemów Cyfrowych, PWN, Warszawa 1972.





## SYSTEMY OPERACYJNE JS EMC

### 1. Wprowadzenie

Środki techniczne JS EMC będą w pierwszej kolejności wyposażone w dwa Systemy Operacyjne:

- DOS/JS, Dyskowy System Operacyjny,
- OS/JS Uogólniony Operacyjny System.

Oba te systemy są przeznaczone do pracy potokowej. Stosowanie ich w zależności od pojemności pamięci przedstawia tabela 1.

Celem niniejszego artykułu jest dostarczenie wstępnych informacji o systemach operacyjnych dla komputera JS, które będą stanowiły podstawowe oprogramowanie dostarczone użytkownikom komputera JS. Podane tu informacje powinny być wystarczające do prowadzenia dyskusji na temat:

- jak zorganizować system eksploatacji korzystając z usług systemu DOS/JS lub OS/JS,
- jak zorganizować prace projektowo-programistyczne w danym ośrodku bazując na systemie DOS/JS lub OS/JS.

Na wstępie należy podać kilka wyjaśnień dotyczących zależności między systemami DOS/JS i OS/JS. Praktycznie, nie ma między nimi wymienności ani programowej, ani informacyjnej. Co prawda istnieje możliwość pracy na takim podzbiórce DOS-u, aby później można było opracowane w DOS-ie programy oraz dane użytkować pod nadzorem systemu OS, ale jest to możliwość teoretyczna. Dla przykładu, jeśli byśmy chcieli tak pracować na poziomie języka Assembler /odpowiednika, czy też raczej "krewniaka" PLAN-u dla

Tabela 1

Pojemność pamięci we/wy Model emc	64kb	128kb	256kb	512kb	1024kb
R-20	DOS	DOS OS-PCP	DOS OS-PCP OS-MFT		
R-30		DOS OS-PCP	DOS OS-PCP OS-MFT	OS-PCP OS-MFT OS-MVT	
R-40			DOS OS-PCP OS-MFT	OS-PCP OS-MFT OS-MVT	OS-PCP OS-MFT OS-MVT
R-50			DOS OS-PCP OS-MFT	OS-PCP OS-MFT OS-MVT	OS-PCP OS-MFT OS-MVT



komputera 1304/, to wybrany podzbiór tego języka nie zawiera żadnych makrorozkazów, a więc brak jest współpracy z urządzeniami we-wy.

Warto zwrócić na to uwagę na wstępie, ponieważ już w początkowej fazie projektowania eksploatacji komputera JS użytkownik powinien zdecydować się w jakim Systemie Operacyjnym będzie pracował.

Wydaje się, że nie jest możliwe, aby można było w początkowej fazie eksploatacji komputera prowadzić prace w dwu systemach. O wprowadzeniu do eksploatacji drugiego systemu można myśleć nie wcześniej niż po roku od czasu rozpoczęcia eksploatacji.

## 2. System operacyjny

### 2.1. Pojęcia podstawowe

System Operacyjny w pojęciu JS jest to zbiór typowych środków programowych o dużym stopniu uniwersalizacji ze względu na zakres zastosowań. Do Systemu Operacyjnego nie wchodzi pakiety zastosowań, ponieważ stanowią niejako same system operacyjny zorientowany pod kątem określonych zastosowań.

System operacyjny pozwala programiście na:

- dzielenie danego zadania na części, programowanie każdej części w języku najlepiej odpowiadającym dla zaprogramowania danej części, a następnie łączenie części w gotowy program;
- umieszczenie gotowego programu w bibliotece systemu i wywołanie go z biblioteki przez nazwę;
- dzielenie dużego programu na segmenty, które podczas wykonywania mogą się na siebie nakładać, co pozwala na zaoszczędzenie pamięci wewnętrznej maszyny;
- automatyzację procesu uruchamiania programu;
- posługiwanie się standardowymi procedurami we-wy dla pracy z danymi,
- wykonywanie programu nie bezpośrednio po jego przetłumaczeniu;
- przechowywanie wyników pośrednich procesu translacji w celu dalszego ich wykorzystania.

System DOS/JS pozwala operatorowi na dość wygodne i zautomatyzowane sterowanie procesem obliczeniowym, ograniczając działalność operatora do koniecznego minimum.

System podaje konserwatorom /zespółom technicznym/ niezbędne informacje do prowadzenia bieżącej kontroli stanu maszyny. System DOS/JS składa się z programów sterujących /planowanie pracy, sterowanie wykonaniem programu i sterowanie danymi/ oraz przetwarzających programów /translatory, programy sterujące/.

Aby maszyna wykonała dowolną pracę, System Operacyjny powinien otrzymać z zewnątrz pracę. Z punktu widzenia organizacji pracy maszyny praca /job/ jest podstawową jednostką rozwiązywanego problemu. Dane zadanie nie zależy od innych zadań i nie może wpływać na inne zadanie, co pozwala na równoległą pracę. Praca jest opisana przy pomocy zdań języka sterującego pracą. Kilka prac tworzy pakiet /potok/. Dane zadanie z punktu widzenia programisty dzieli się na kroki /step/.

System Operacyjny, a dokładniej: jego programy sterujące organizują przyjęcie pracy, ich kontrolę, przygotowanie żądanych programów do wykonania, uruchomienie programu i automatyczne przejście do następnej pracy. Wszystkie te czynności nazywa się zarządzaniem pracami.

Za każdym razem, kiedy program rozpoznaje kolejny etap pracy, przyjmuje go jako zadanie /zlecenie/. Aby zlecenie mogło być wykonane, system przydziela mu niezbędne środki realizacyjne: pamięć wewnętrzną i zewnętrzną, urządzenie wejścia-wyjścia, czas realizacji procesu itp. Zlecenie znajduje się pod bezpośrednią kontrolą supervisora. Zlecenie może być realizowane w reżimie jedno- lub wieloprogramowym. Reżim jednoprogramowy - całe wyposażenie jest do dyspozycji programu.

### 2.2. System operacyjny DOS/JS - wprowadzenie

DOS dopuszcza wykonanie trzech zleceń równocześnie - jest to wieloprogramowość z ustaloną ilością /do trzech/ programów.

Wieloprogramowość z ustaloną ilością programów charakteryzuje się tym, że obszar pamięci operacyjnej jest rozdzielony na stałą ilość części. Wielkość pamięci przydzielona na wykonanie poszczególnych programów jest określana przy inicjacji systemu.

Przy tak przyjętej strategii wieloprogramowości rozróżnia się programy problemowe dwóch rodzajów: programy drugoplanowe tzw. BG-programy i programy pierwszoplanowe, tzw. F-programy.

BG-programy są uruchamiane przez Job Control zgodnie z opisem zadań /a więc jedynie w przetwarzaniu grupowym/, przy czym w jednym kroku zadania uruchamiany jest jeden program /jedno- lub wielofazowy/.

F-programy mogą być uruchamiane pojedynczo przez operatora, który podaje odpowiednie informacje systemowemu inicjatorowi F-programów, lub też w przypadku, gdy zestaw komputera jest odpowiedni /obszar pierwszoplanowy ma pojemność co naj-



mniej 10K/ i do dyspozycji jest dostateczna liczba urządzeń wejścia-wyjścia - przez Job Control.

Przy odpowiednim zestawie istnieją więc możliwości przetwarzania grupowego również i w pierwszoplanowych obszarach programowych. Rodzaj przetwarzania w pierwszoplanowych obszarach programowych określa się podczas generowania systemu.

BG-programy i F-programy "zaczynają się" i "kończą" niezależnie od siebie i są logicznie niezależne. System może obsługiwać jednocześnie program drugoplanowy i jeden lub dwa programy pierwszoplanowe.

Priorytety nadawane są programom automatycznie z chwilą ich wprowadzenia do pamięci operacyjnej i są determinowane przez obszary, w których te programy będą pracować. Najwyższy priorytet nadawany jest programowi znajdującemu się w obszarze F1. Niższy priorytet ma program z F2, a najniższy program z BG. Po obsłużeniu przerwania Supervisor przekazuje sterowanie temu z programów z będących w stanie gotowości do pracy, który ma najwyższy priorytet. Jeżeli żaden z programów nie jest w stanie gotowości /na przykład wszystkie programy czekają na zakończenie rozpoczęcia translacji/, to system przechodzi w stan czekania na przerwanie.

Aby przetwarzać dane np. przy pomocy standardowych procedur we-wy należy zapewnić przechowywanie, wyszukiwanie i przesyłanie tych danych. Dane są zorganizowane w dokumenty logiczne, które stanowią jednostki informacji. Wszystkie operacje przetwarzania wykonywane są na podstawie programów sterowania danymi DOS/JS. Zbiór wszystkich tych programów tworzy System Sterowania Wejściem-Wyjściem. Zbiory dokumentów są organizowane w pliki. Plik rozmieszczony na konkretnym nośniku nazywa się Voluminem. Jeden Volumin może zawierać wiele plików. Charakterystyki plików są przechowywane w etykietach pliku. Podobne dane zapisywane są w etykietach Voluminu.

Maszyny JS dopuszczają podłączenie szerokiej klasy urządzeń zewnętrznych. Dla zagwarantowania niezależności programisty od konkretnego urządzenia przyjęto metodę tzw.

logicznych urządzeń. Po prostu zamiast nazw fizycznych używa się w programie standardowych nazw symbolicznych tych urządzeń. Zaś przed wykonaniem programu w języku "sterującym" przyporządkowuje się konkretnemu urządzeniu logicznemu jego charakterystykę fizyczną.

Nazwa symboliczna ma postać SYSXXX, gdzie XXX wyraża dowolny znak lub liczbę. Część nazw jest zastrzeżonych dla systemowych urządzeń logicznych.

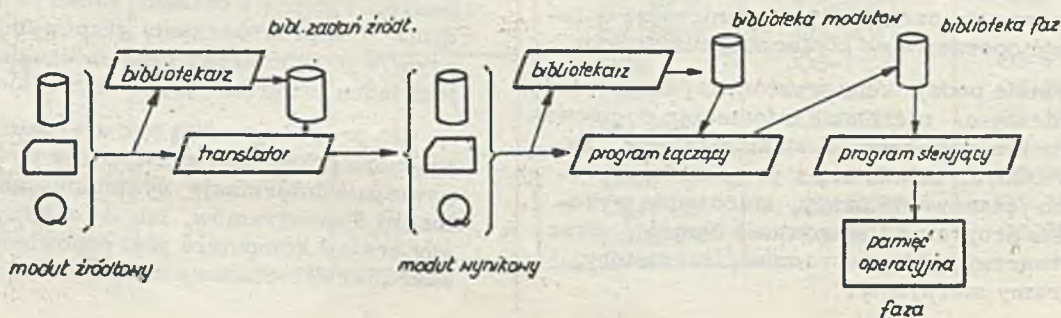
### 2.3. Zawartość systemu DOS/JS i etapy przetwarzania

Poniższy rysunek zawiera podstawowe elementy systemu DOS/JS.

Program sterujący	Programy przetwarzające
IPL	Translatory
	Assembler
Supervisor	FORTRAN podstawowy
	RPG
Job-Control	PL/1
System sterowania wejściem-wyjściem:	Programy serwisowe:
- poziom fizyczny	Program łączący /linkage Editor/
- poziom logiczny	Bibliotekarz /Librarian/
	Programy sortowania łączenia Sort/Merge
	Systemowe programy pomocnicze /Utilities/
	Autotest - program do automatyzacji uruchamiania programów
	Programy problemowe:
	Programy napisane przez użytkowników

Nazwy poszczególnych elementów przedstawione na rysunku mają podobne znaczenie jak w innych Systemach Operacyjnych np. w Systemie DOS dla IBM 360.

Etapy przetwarzania przedstawia rys. 1.



Rys. 1



## 2.4. System Operacyjny OS/JS

### Wprowadzenie

Z Systemu OS, w zależności od potrzeb, użytkownik może wygenerować trzy wersje systemu: OS-PCP, OS-MVT i OS-MFT. Poniżej omawiamy każdą z wersji systemu OS. Należy zaznaczyć, że określone zadanie wykonane na danym zestawie środków technicznych w jednym z w/w systemów, będzie dawało takie same rezultaty w innym systemie.

Wymienność programów pomiędzy różnymi wersjami systemu przedstawia rys. 2. /program pracujący w PCP pracuje w MFT i MVT itd./

System PCP jest wersją Systemu OS/JS, w której można pracować już na konfiguracjach ze 128 kb pamięci operacyjnej. System PCP wczytuje strumień informacji składający się z prac uporządkowanych sekwencyjnie oraz inicjuje i nadzoruje wykonywanie jednej pracy w danym przedziale czasowym.



Rys. 2

Jest to, inaczej mówiąc, wersja systemu pracująca w reżimie jednoprogramowym.

Rys. 3 przedstawia sposób pracy Systemu PCP.

Dla współpracy z urządzeniami zewnętrznymi w PCP jest zespół programów, nazywanych programami metod dostępu do urządzeń zewnętrznych. Służą one zarówno dla użytkownika do organizacji współpracy z urządzeniami zewnętrznymi jak i do realizacji różnych funkcji samego Systemu, takich jak np. czytanie pracy, wypisywanie pracy itp.

System MFT może nadzorować wykonanie więcej niż jednego zadania w danym czasie. System ten ma możliwość nadzorowania stałej ilości zadań w danym przedziale czasowym. Możliwość równoczesnego przetwarzania czyli tzw. wieloprogramowość w tym systemie przedstawia rys. 4.

Pamięć operacyjna komputera dostępna dla programów jest dzielona przez operatora

podczas inicjacji Systemu na części zwane "porcjami". Im większy jest adres pamięci dla danej porcji, tym większy jest jej priorytet. Z porcjami związane są klasy. Każdemu z kroków pracy musi być przyporządkowana przez użytkownika klasa. Klasa wiąże dany krok pracy z daną porcją.

System MVT może przetwarzać jednocześnie do 15 niezależnych prac. Zasady pracy Systemu MVT przedstawione są schematycznie na rys. 5.

Pamięć operacyjna dostępna dla przetwarzania programów użytkowych jest przydzielona poszczególnym pracom dynamicznie po zakończeniu danej pracy.

Schemat opracowywania programu przez System Operacyjny OS/JS

Proces opracowywania przez System Operacyjny danego programu użytkowego, napisanego w jednym z dostępnych języków, przebiega wg rys. 6.

### Zawartość systemu

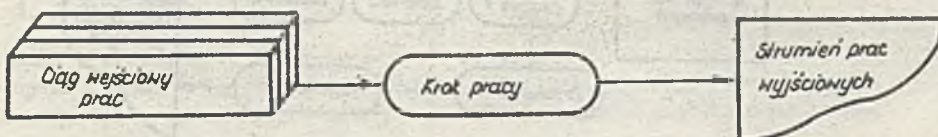
Poniższa tabela zawiera dane dotyczące zawartości Systemu OS/JS

Program sterujący	Programy przetwarzające		
	Translatory	Programy serwisowe	Programy problemowe
Zarządzanie pracami	Assembler, FORTRAN	Program łączący, ładowacz,	Programy napisane przez użytkowników
Zarządzanie zadaniami	ALGOL-60, COBOL-ANS	Sort-Merge, TESTRAN, Systemowe programy pomocnicze, Programy pomocnicze dla zbiorów danych, Niezależne programy pomocnicze	
Zarządzanie danymi	PL/1F, RPG		

Czytanie i interpretacja jednej pracy lub kroków pracy z jednego strumienia wejściowego

Przetwarzanie jednego z kroków pracy

Wytwarzanie pracy wyjściowej na jedno urządzenie



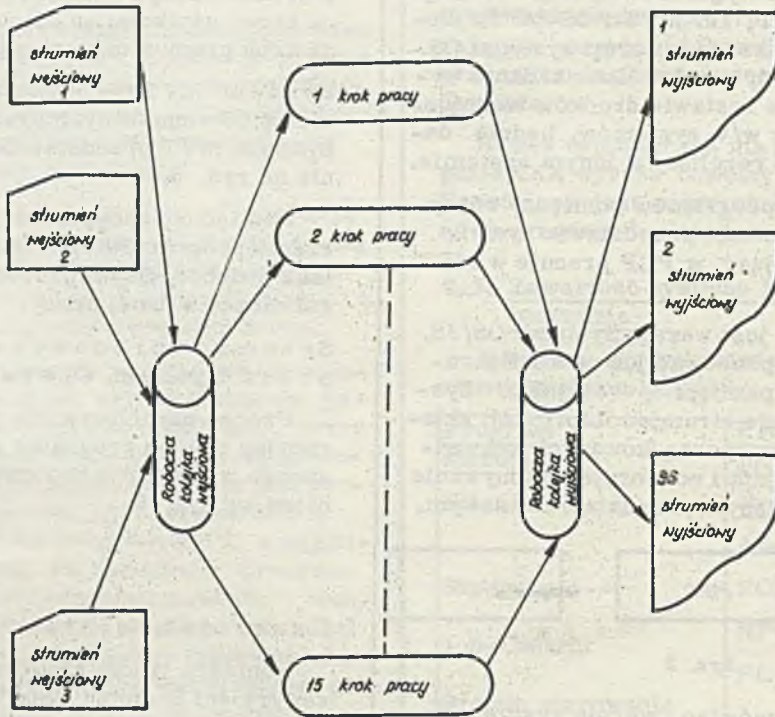
Rys. 3



Czytanie i interpretacja prac z 1 do 3 wejściowego strumienia

Przetwarzanie od 1 do 15 prac

Wygenerowanie od 1 do 36 prac wyjściowych

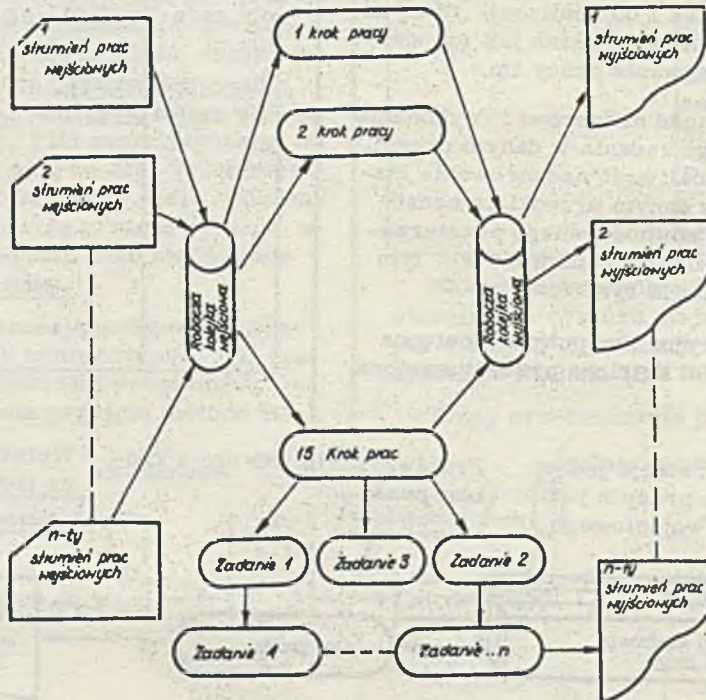


Rys. 4

Czytanie i interpretowanie prac z ciągu wejściowego

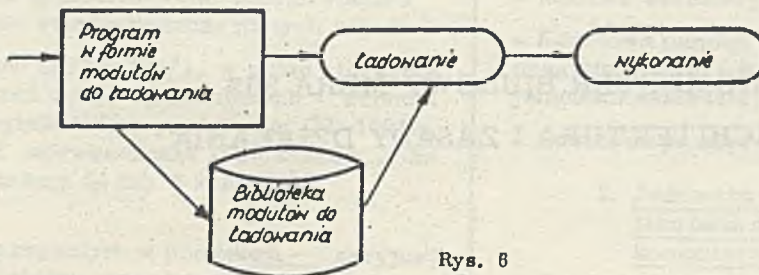
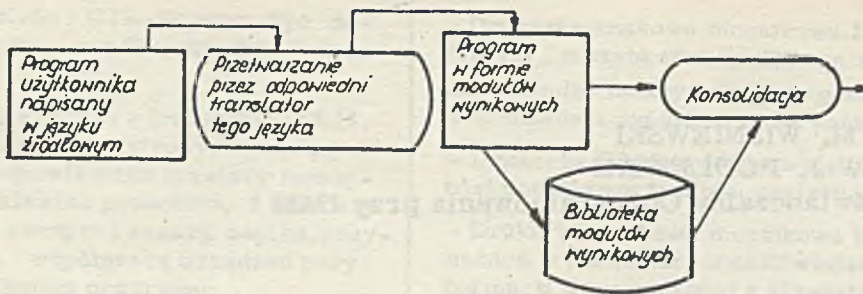
Przetwarzanie od 1 do 15 prac

Wygenerowanie prac wyjściowych na dowolną ilość urządzeń



Rys. 5





Rys. 6

### 3. Porównanie systemów

Poniżej zamieszczamy krótkie porównanie omawianych systemów

programistów z OBR MERA-ELWRO kierowanych przez Z-cę Głównego Konstruktora JS EMC mgr St. Lepetowa, za co niniejszym dziękuję.

cecha systemu / nazwa systemu	DOS/JS	OS/JS
wieloprogramowość	Do 3 programów równocześnie ze stałym podziałem sprzętu i pamięci pomiędzy programy	Do 16 programów równocześnie z możliwością pracy w reżimie dynamicznego podziału sprzętu oraz pamięci
teleprzetwarzanie	Brak wyspecjalizowanych środków programowych do organizowania systemu teleprzetwarzania	Bogaty zestaw środków programowych do teleprzetwarzania już na najniższym poziomie programowania metody dostępu TCAM
GENEROWALNOŚĆ i adoptowalność systemu	Proces generowania dotyczy tylko wyprodukowania systemu dla zadanej konfiguracji sprzętowej	Proces generowania pozwala nie tylko na otrzymanie systemu dla konkretnej konfiguracji, ale również wyspecjalizowanego z punktu widzenia przewidywanych zastosowań aż do poziomu transl.
organizacja współpracy z urządzeniami zewnętrznymi	Proste środki programowe /access method/ w postaci makrorozkazów ułatwiających współpracę z urządzeniami. Organizacja współpracy zależy od programisty	Pełny zestaw środków programu /data management/ zwalniający programistę i projektanta od organizacji współpracy z urządzeniem zewnętrznym.
zasady konstrukcji systemu	Istnieje standard modułu lecz system nie jest modułowy, a raczej monolityczny. Wymiana jakiegokolwiek części systemu /np. w celu włączenia nowego urządzenia/ wymaga głębokiej jego znajomości	System napisany zgodnie z zasadami modularnego programowania, w związku z tym wystarczy znajomość standardu modułu aby wymienić dowolny moduł systemu na inny
ZESTAWY PAKIETÓW	Ilość pakietów rozszerzająca możliwości systemu. Mała stosunkowo ilość pakietów problemowo zorientowanych	Około 3-4 razy większa ilość pakietów problemowo zorientowanych niż w DOS. W zasadzie wszystkie pakiety posiadają języki problemowo zorientowane jak metodą opr. zagadnień. do rozw.

### 4. Zakończenie

Przy omawianiu w niniejszym artykule zagadnień związanych z Systemem Operacyjnym DOS/JS autor korzystał z opracowań grupy

Jednocześnie autor pragnie podziękować: mgr J. Swianiewiczowi, mgr W. Kozłowskiemu i mgr inż. St. Niewolskiemu z ZDO przy IMM za udostępnione materiały.



ANDRZEJ M. WIŚNIEWSKI  
STANISŁAW J. PODLEWSKI  
Zakład Doświadczalny Oprogramowania przy IMM

## KOMPUTER BIUROWY MERA 302 ARCHITEKTURA I ZASADY DZIAŁANIA

### 1. Wprowadzenie

Niniejsze opracowanie powstało w oparciu o wyniki prac zespołu Zakładu Doświadczalnego Oprogramowania przy Instytucie Maszyn Matematycznych. W skład zespołu wchodzi: mgr J. Walasek, mgr D. Kosecka, mgr S. Sawicki, mgr Z. Wrotek. W szczególności wykorzystano dwie publikacje w/w autorów:

- "Komputer biurowy MERA 302" /wyd. ZDO przy IMM, 1973/.
- "Komputer biurowy MERA 302 - instrukcja operatorska" /wyd. ZDO przy IMM, 1973/.

Można tam również znaleźć szczegółowe informacje dotyczące zasad działania i zakresu zastosowań komputera biurowego MERA 302.

Prace nad komputerem biurowym MERA 302 są prowadzone w Zakładzie Doświadczalnym Oprogramowania przy Instytucie Maszyn Matematycznych na zasadzie współpracy z Zakładami Wytwórczymi Przyrządów Pomiarowych "ERA", które są producentem sprzętu i jego dostawcą.

### 1. Podstawowe dane

#### o sprzęcie MERA 302

Produkowane obecnie urządzenia wchodzące w skład systemu MERA 302 mogą być dostarczane w następujących konfiguracjach:

1/ Zestaw podstawowy, w skład którego wchodzi:

- Jednostka centralna MKO8 /minikomputer Momik 8b/, wyposażona w ferrytową pamięć operacyjną o pojemności 8 K słów 8-bitowych i czasie cyklu 1,8 us. Długość rozkazu wy-

nosi 8 lub 16 bitów, lista rozkazów zawiera 32 podstawowe instrukcje logiczne, arytmetyczne i sterujące. Arytmetyka: binarna, uzupełnieniowa, równoległa, na 8 bitach. MKO8 może być wyposażony w dwa typy kanałów - kanał arytmometru /do którego można dołączyć do 12 urządzeń wejścia-wyjścia/ i kanał multipleksera /opcjonalny/, wyposażony w 16 podkanałów. Minikomputer MKO8 ma rozbudowany system przerw w pięciu poziomach, z których trzy przyporządkowano przerwaniom zewnętrznym. Na każdym poziomie może występować do 32 przerw zewnętrznych priorytetowych;

- Jednostka sterująca urządzeniami peryferyjnymi PUO2, wyposażona we własne układy zasilania, umożliwiającą montaż do 12 pakietów jednostek sterujących urządzeniami wejścia-wyjścia;

- Elektryczna maszyna do pisania MPO1 /Facit 3851/, o szybkości pisania 15 znaków na sekundę, będąca jednocześnie urządzeniem wejścia /klawiatura/ i wyjścia /drukarka/. Maszyna wyposażona jest w pełen repertuar znaków alfanumerycznych /w tym wszystkie litery alfabetu polskiego/ i funkcyjnych, wydruk może być wykonywany w dwóch kolorach;

- Czytnik taśmy papierowej i kart brzeźnie perforowanych CTK-50, o szybkości działania 35 znaków na sekundę odczytywanych z taśmy papierowej 5- lub 8-kanałowej albo kart brzeźnie perforowanych;

- Dziurkarka taśmy papierowej DTK-50, o szybkości wyprowadzania 35 znaków w ciągu sekundy na taśmę papierową 5- lub 8-kanałową, albo na karty brzeźnie perforowane



/urządzenia DTK-50 i CTK-50 mogą być dołączane przez jedną, wspólną dla obu, jednostkę sterującą/.

- Klawiatura numeryczno - funkcyjna KL11, zawierająca 10 klawiszy cyfrowych /będących powtórzeniem odpowiednich klawiszy maszyny do pisania/, klawisz przecinka, 2 klawisze znaku, klawisze akceptu i kasacji zapisu, przyciski sterowania współpracą urządzeń peryferyjnych i działaniem programu;

- Konsola operatorska, wyposażona w przełączniki zasilania i startu - stopu oraz w klucze i lampki sygnalizacyjne, umożliwiające odczytywanie stanów poszczególnych rejestrów

2/ Zestaw MERA 302/1, w skład którego - obok urządzeń wyżej wymienionych - wchodzi ponadto czytnik taśmy papierowej CT-1001A, o szybkości wprowadzania 1000 znaków na sekundę z taśmy 5- lub 8-kanalowej.

W roku przyszłym w produkcji seryjnej znajdą się kolejne urządzenia peryferyjne, o które będzie mogła być rozbudowywana konfiguracja komputera biurowego MERA 302. Będą to:

- Drukarka znakowa mozaikowa DZM-180 /DZ11/, o szybkości wypisywania 180 znaków na sekundę, maksymalnej długości wiersza 158 znaków i repertuarze 64 znaków;

- Drukarka DZ12, różniącą się od poprzedniej podwójnym traktem papieru,

- Drukarka znakowa mozaikowa DZ21, wyposażona w klawiaturę umożliwiającą druk informacji wprowadzanej z klawiatury i wprowadzanie informacji z klawiatury do minikomputera;

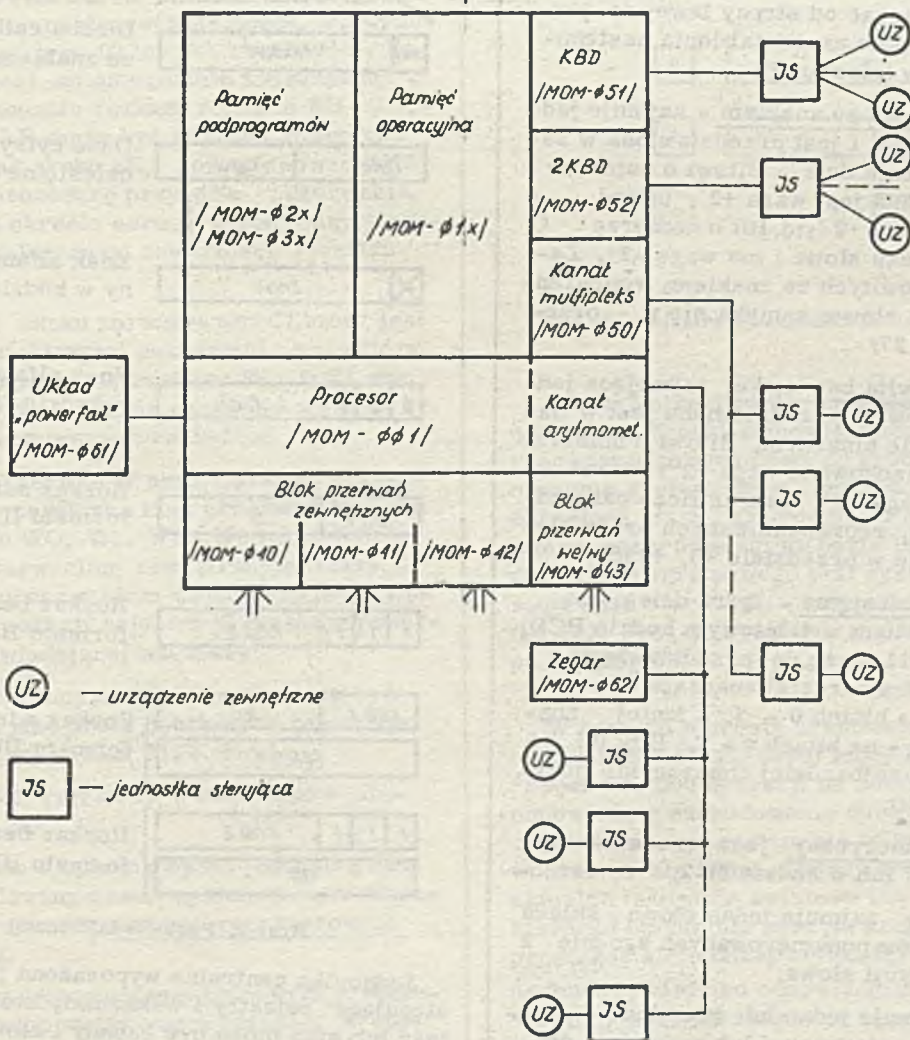
- Monitor ekranowy alfanumeryczny MA01;

- Kasetowa pamięć dyskowa PD10 z jednym dyskiem stałym i jednym zmiennym, o pojemności całkowitej około 6 M bajtów,

- Kasetowa pamięć taśmowa TKO1.

2. Jednostka centralna MKO8 jako baza dla budowy systemu komputera biurowego MERA 302

Jednostka centralna MKO8 /minikomputer Momik 8b/ jest modularnym systemem cyfrowym, w skład którego wchodzi:



Rys. 1. Schemat blokowy systemu MOMIK 8b



- Procesor, podstawowy blok przetwarzania informacji i sterowania pracą systemu, obejmujący także kanał arytmometru;
- Pamięć operacyjna, służąca do przechowywania rozkazów i danych dla programów wykonywanych przez procesor;
- Blok przerwań wejścia-wyjścia, umożliwiający przyjmowanie sygnałów przerwań z 32 urządzeń wejścia-wyjścia;
- Kanał multipleksera, który służy do blokowego przesyłania informacji między pamięcią operacyjną i urządzeniami wejścia-wyjścia w 16 podkanałach;
- Układ zabezpieczenia przed zanikiem zasilania /power fail/, pozwalający na przechowanie stanu systemu przy zaniku napięcia zasilającego i na ponowne wystartowanie programu po powrocie napięć zasilających do wartości nominalnych. Architekturę jednostki centralnej ilustruje rysunek 1.

Momik 8b jest maszyną cyfrową pracującą w systemie binarnym. Podstawową jednostką informacji jest słowo składające się z 8 bitów. Bity /pozycje/ słowa są ponumerowane umownie od 0 do 7, licząc od strony lewej do prawej. Słowo służy do przedstawiania następujących rodzajów informacji:

- Liczba całkowita ze znakiem - zajmuje jedno słowo 8-bitowe i jest przedstawiona w zapisie "uzupełnienie do 2". Bitowi o numerze 7 przyporządkowana jest waga  $+2^7$ , bitowi o numerze 6 - waga  $+2^6$  itd, Bit o numerze 0 jest pozycją znaku słowa i ma wagę  $-2^7$ . Zakres liczb całkowitych ze znakiem, reprezentowanych przez słowo; zamyka się w przedziale  $-128, +127$ ;

- Liczba całkowita bez znaku, zajmująca jedno słowo 8-bitowe, przedstawiana jest w naturalnym zapisie binarnym. Bitowi o numerze 7 jest przyporządkowana waga  $+2^0$ , bitowi o numerze 0 - waga  $+2^7$ . Zakres liczb całkowitych bez znaku, reprezentowanych przez słowo, zawiera się w przedziale  $+0, +255$ .

- Dwie cyfry dziesiętne - cyfra dziesiętna jest przedstawiona w 4-bitowym kodzie BCD. Kody 1010 i 1011 służą do przedstawienia znaku "-". Cyfra bardziej znacząca jest umieszczona na bitach 0 + 3, a mniej znacząca lub znak - na bitach 4 + 7. Bity 0 i 4 są odpowiednio najbardziej znaczącymi pozycjami kodu BCD;

- Znak alfanumeryczny jest przedstawiony w kodzie ISO-7 lub w kodzie M-2;

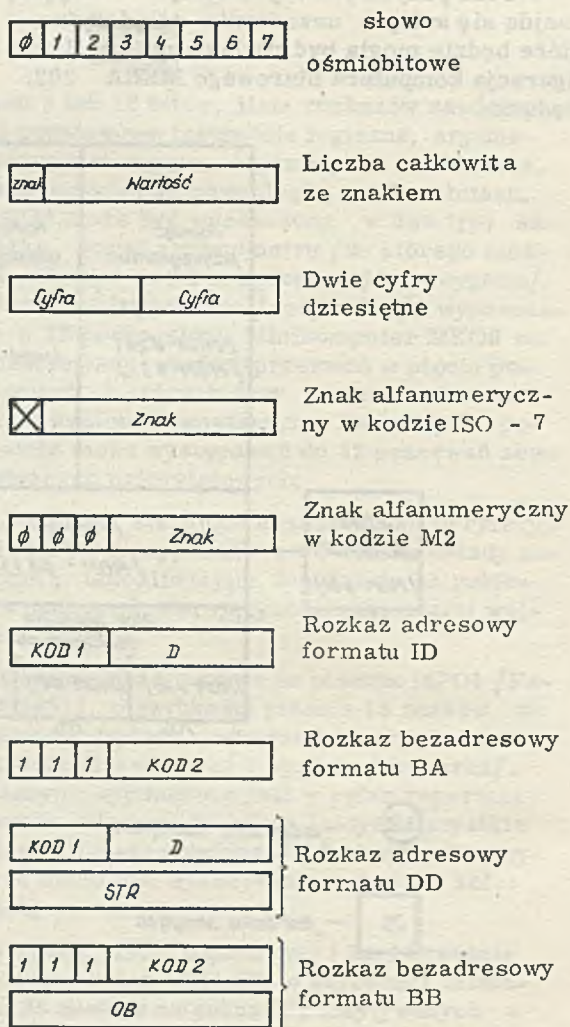
- Znak binarny zajmuje jedno słowo i składa się z ciągu bitów ponumerowanych zgodnie z numerami pozycji słowa;

- Rozkaz zajmuje jedno lub dwa słowa 8-bitowe /może być pojedynczej lub podwójnej długości/. Poszczególne pola w słowach rozka-

zowych mają następujące znaczenie:

- KOD 1 - /pozycje 0 + 2/ jest trzybitową częścią operacyjną, pozwalającą na rozróżnienie siedmiu rozkazów adresowych /kody 000 + 110/ i grupy rozkazów bezadresowych /kod 111/;
- KOD 2 - /pozycje 3 + 7/ jest częścią operacyjną rozkazów bezadresowych,
- D - /pozycje 3 + 7/ jest przesunięciem określającym położenie operandu rozkazu adresowego w obrębie strony określonej zawartością rejestru S lub słowa STR,
- STR - określa numer strony dla rozkazu; adresowego SK /kod 100/;
- OB - jest operandem bezpośrednim dla niektórych rozkazów bezadresowych.

Postacie informacji przedstawia rys. 2.



Rys. 2. Postacie informacji

Jednostka centralna wyposażona jest w następujące rejestry i wskaźniki, których wartość lub stan może być badany i zmieniany przez program:



- Rejestr akumulatora A, o długości 8 bitów, służy do przechowywania operandów rozkazów. Dla operacji dwuargumentowych rejestr A jest źródłem jednego z operandów i miejscem umieszczenia wyniku działania rozkazu;

- Wskaźnik przeniesienia P jest jednobitowym przedłużeniem rejestru akumulatora, służy do rejestrowania nadmiarów występujących przy operacjach na zawartościach akumulatora;

- Rejestr strony S, o długości 8 bitów, zawiera numer strony w pamięci operacyjnej, w której znajdują się operandy rozkazów. Położenie operandu w obrębie strony określa 5-bitowe przesunięcie D rozkazu;

- Wskaźnik strony zerowej Z, który określa sposób tworzenia adresu w pamięci operacyjnej. Gdy stan wskaźnika Z jest równy 1, adres operandu określa przesunięcie D rozkazu w obrębie strony zerowej. Jeśli wskaźnik Z jest równy 0, przy tworzeniu adresu operandu uwzględniona jest zawartość rejestru strony S;

- Licznik rozkazów LR, będący rejestrem 13-bitowym, którego zawartość po wykonaniu rozkazu określa adres rozkazu mającego być wykonanym w następnej kolejności. Po wykonaniu rozkazu typu ID lub BA zawartość rejestru LR jest automatycznie zwiększana o 1, a po wykonaniu rozkazu formatu BB o 2. Zawartość LR może być również zmieniana przez rozkaz skoku SK, rozkaz powrotu PW i PO oraz procedurę przyjęcia przerwania. Rejestr LR określa adresy wykonywanych rozkazów niezależnie od zawartości rejestru strony S;

- Wskaźnik skoku warunkowego CI, który jest ustawiany niektórymi rozkazami, a który warunkuje wykonanie rozkazu SK. Jeśli stan CI jest równy 1, rozkaz skoku nie zmienia sekwencji wykonywanych rozkazów;

- Rejestr przerwania W umożliwiający maskowanie poszczególnych klas przerwania Momika 8b. Pozycje W0, W1, W2 i W3 odpowiadają kolejno przerwaniom zewnętrznym klasy 0, 1 i 2 oraz przerwaniom wejścia-wyjścia. Zerowy stan pozycji rejestru W maskuje przerwania odpowiadające mu klasy;

- Pamięć operacyjna Momika 8b może być traktowana jako zbiór 8-bitowych rejestrów. Pojemność jej wynosi 8192 słowa.

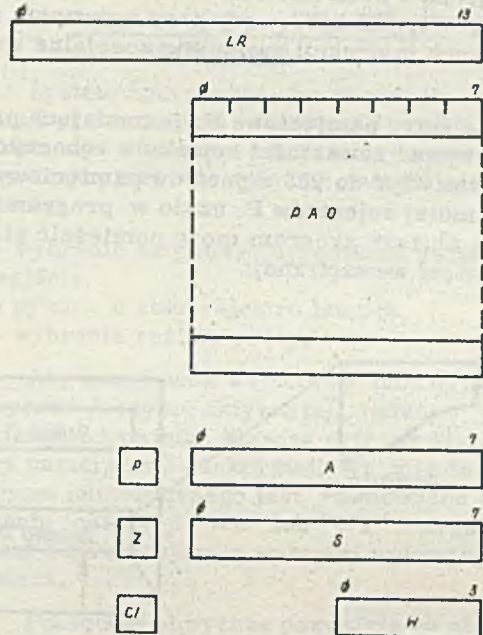
Wielkości rejestrów i wskaźników ilustruje rys. 3.

W oparciu o opisaną wyżej jednostkę centralną zbudowany został system programowania, który nazwano komputerem biurowym MERA 302.

Praca nad komputerem biurowym przebiegała w dwóch podstawowych etapach. W etapie pierwszym opracowano:

- Prosty język symboliczny SYWIK. Rozkaz tego języka składa się z mnemotechnicznej części operacyjnej, odpowiadającej jednemu rozkazowi MOMIKA 8b i adresu w formie symbolicznej etykiety;

- Środki programowe do automatyzacji procesu uruchamiania systemów, pozwalające na pracę w reżimie konwersacyjnym, z możliwością wypisywania stanów rejestrów i zawartości poszczególnych pól pamięci po wykonaniu rozkazu.



Rys. 3. Rejestry i wskaźniki m. c. MOMIK 8b

W wyniku realizacji etapu pierwszego, otrzymano środki stanowiące niezbędne oprzyrządowanie programowe do produkcji oprogramowania systemowego, zorientowanego na określoną klasę zagadnienia. Pierwszym systemem opracowanym na bazie tego oprzyrządowania programowego jest system automatyzacji prac biurowych - komputer biurowy MERA 302. Jest on wynikiem realizacji prac etapu drugiego.

W oparciu o prostą architektonicznie jednostkę centralną o dużej mocy obliczeniowej /rzędu 300 000 operacji na sekundę/, poprzez odpowiednie nadbudowanie oprogramowania, otrzymano więc system nie tylko w pełni nowoczesny, ale i tani. Należy podkreślić, że aktualne tendencje światowe idą właśnie w kierunku budowania sprzętu architektonicznie prostego, ale o dużej szybkości działania i nadbudowywania go odpowiednim oprogramowaniem, zorientowanym na określone klasy zastosowań.



#### 4. Komputer biurowy - zasady działania

Pamięć komputera biurowego MERA 302 zawiera następujące obiekty:

- 16 rejestrów roboczych R, pełniących funkcje akumulatorów. Rejestr roboczy może zawierać jedną liczbę lub fragment tekstu. Wszystkie operacje wykonywane są w rejestrach roboczych. Poprzez te rejestry dokonuje się również wprowadzania i wyprowadzania danych;

- Rejestr warunku C automatycznie otrzymuje jedną z czterech wartości, oznaczanych: WO, W1, W2 i W3, gdy przy wykonaniu niektórych instrukcji zaistnieją specjalne warunki;

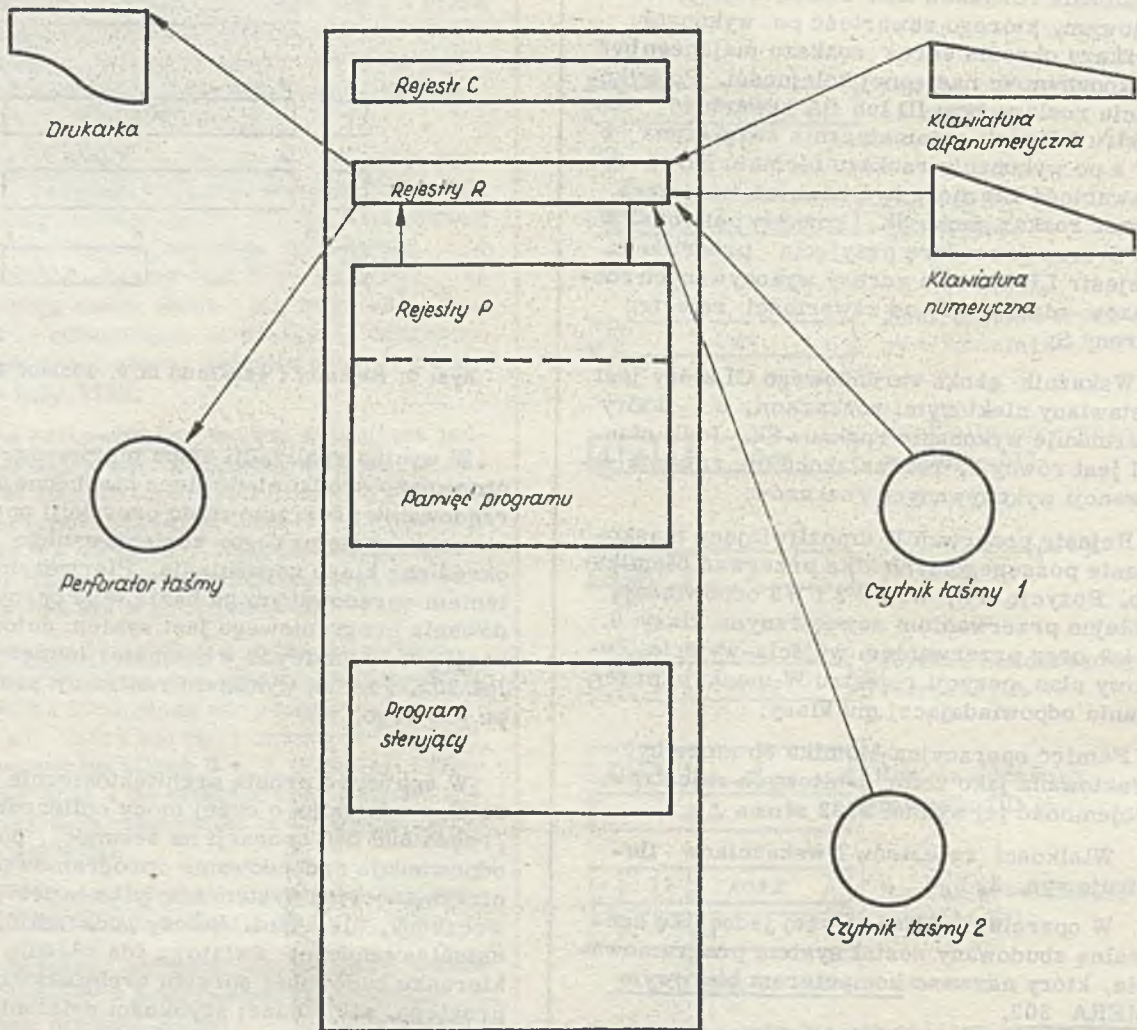
- Rejestry pamięciowe P, pozwalające przechowywać zawartości rejestrów roboczych. Można użyć do 256 rejestrów pamięciowych. Im mniej rejestrów P użyto w programie, tym dłuższy program może pomieścić się w pamięci wewnętrznej;

- Pamięć programu zawierającą poszczególne instrukcje programu użytkowego. Instrukcje te działają na wymienionych wyżej rejestrach. Pość instrukcji mieszcząca się w pamięci zależy od ich charakteru. Orientacyjnie można przyjąć, że jeden program liczy może około 1000 instrukcji;

- Program sterujący, który nadzoruje wykonanie programu użytkowego.

Organizację pamięci przedstawiono na rys. 4.

Program w postaci symbolicznej można wprowadzać do komputera biurowego z czytnika lub z klawiatury maszyny do pisania. Przy wprowadzaniu można żądać drukowania tekstu programu, w celu sprawdzenia poprawności zapisu i wczytywania. Program wprowadzony do komputera biurowego może być wykonywany w reżimie ciągłym lub krokowym. Wykonywanie krokowe stosuje się przede wszystkim w reżimie testowania i uruchamiania /debug-



Rys. 4. Organizacja pamięci



ging/ programu. Przy wykonywaniu krokowym operator może zatrzymywać program w zaznaczonych miejscach, wykonywać po jednej instrukcji, wypisywać pośrednie wartości rejestrów oraz zmieniać zawartości rejestrów sterowania. Operacje te może operator wykonywać również za pomocą instrukcji komputera biurowego, wprowadzanych z klawiatury maszyny. Instrukcje wykonują się niezależnie od programu wewnętrznego.

Każdy rejestr składa się z odpowiedniej ilości słów. Jedno słowo może zawierać jeden znak alfanumeryczny lub jedną do dwóch cyfr dziesiętnych.

#### Rejestr R

Rejestr roboczy R złożony jest z 16 słów. Rejestr ten może zawierać liczbę dziesiętną lub fragment tekstu. Jeśli zawiera liczbę, to w jednym słowie znajduje się algebraiczny znak liczby, w jednym informacja o położeniu przecinka dziesiętnego, a pozostałe czternaście słów zajmują cyfry liczby. Pozycje cyfrowe ponumerowane są kolejnymi liczbami od zera do trzynastu, od prawej skrajnej poczynając. Położenie przecinka określa się podając numer pozycji, za którą przecinek się znajduje. Tak zdefiniowane położenie przecinka nazywane jest skalą tej liczby. Jeśli w rejestrze R znajduje się tekst, wówczas kolejne słowa zawierają kolejne znaki tekstu. Znaki tekstu numerowane są od zera do piętnastu, od lewego skrajnego poczynając.

#### Rejestr C

Rejestr warunkowy C składa się z jednego słowa 8-bitowego.

#### Rejestr P

Rejestr pamięciowy P składa się z ośmiu słów. Najczęściej zawartość rejestru R jest zapamiętywana w dwóch kolejnych rejestrach P. W przypadku jednak gdy w rejestrze R znajduje się liczba, można ją zapamiętać w postaci spakowanej w jednym rejestrze P. Jest to możliwe dlatego, że przy zapisie liczby w rejestrze R używa się jedynie połowy słów.

#### Instrukcje

Komputer biurowy zawiera 59 różnego rodzaju instrukcji. Pojedyncza instrukcja składa się z dwuliterowego oznaczenia i symboli rejestrów, do których instrukcja odnosi się, np.:

RD R1, R7; oznacza: do pierwszego rejestru roboczego dodaj siódmy rejestr roboczy,

QY R2, R14; czytaj liczbę do R2, a znak ją ograniczający do rejestru R14,

ZU R3, P200; R3 zapamiętaj w P200 /dwusetnym rejestrze pamięciowym/.

PUR5, R15; wypisz liczbę z R5 według formatu znajdującego się w R15,

DL E29; etykieta - następną instrukcję oznacz symbolem E29,

ST E48; skocz do instrukcji oznaczonej symbolem E48.

Pełen wykaz instrukcji zawarty jest w wykazie.

#### 5. Komputer biurowy - zasady użytkowania

System operacyjny - po włączeniu zasilania i uruchomieniu maszyny - czeka na informacje zadawane z klawiatury numeryczno-funkcyjnej. Mogą to być następujące polecenia:

- wybranie aktywnego urządzenia wejścia lub wyjścia,
- pytanie o stan rejestru lampek,
- wybranie režimu pracy.

Aby urządzenie wejściowe lub wyjściowe wybrać /uczynić aktywnym/, należy przycisnąć odpowiedni klawisz cyfrowy klawiatury numeryczno-funkcyjnej. Wciśnięcie klawisza numerycznego jest równoważne wykonaniu instrukcji "WE" lub "WY" z argumentem odpowiadającym wartości wybranego klawisza.

Polecenia powyższe pozwalają na pisanie programów użytkowych niezależnych od urządzeń wejścia-wyjścia /mogących pracować na dowolnym zestawie tych urządzeń/.

Naciśnięcie klawisza z przecinkiem umożliwia odczytanie stanu rejestru lampek, ładowanego przez instrukcję "ZA". Zawartość rejestru lampek odczytuje się z lampek L, przy wciśniętym kluczu "A".

Klucze funkcyjne P1, P2, P3, P4 służą do wybrania jednego z režimów pracy:

P1 - powoduje przejście do czytania programu symbolicznego podawanego do aktualnie wybranego urządzenia wejściowego;

P2 - powoduje przejście do czytania programu binarnego, podawanego do aktywnego urządzenia wejściowego,

P3 - powoduje przejście do pracy krokowej,

P4 - powoduje przejście do pracy ciągłej programu.

Po znakach P1, P2, P3 i P4 należy zawsze przycisnąć klawisz "A". Po przejściu do jednego z wyżej opisanych režimów pracy powoduje to zgaszenie lampki sygnalizacyjnej "GOTOW".



## 6. Zasady dostaw sprzętu technicznego komputera biurowego

Dostawy sprzętu realizowane są przez Zakłady Wytwórcze Przyrządów Pomiarowych "ERA", według kolejności wpływu zamówień.

Sprzęt objęty jest serwisem, w ciągu pierwszych 12 miesięcy - gwarancyjnym, a po upływie okresu gwarancyjnego - odpłatnym, prowadzonym przez producenta. Zakres serwisu obejmuje pełną obsługę techniczną, zarówno bieżącą konserwację jak i usuwanie ewentualnych usterek, mogących wystąpić w trakcie eksploatacji użytkowej.

Producent sprzętu prowadzi szkolenie pracowników obsługi eksploatacyjnej komputera biurowego w zakresie:

- Obsługi operatorskiej - osoby przeszkolone w tym zakresie otrzymują zaświadczenia uprawniające do obsługi programów użytkowych. Posiadanie przeszkolonej obsługi operatorskiej jest warunkiem objęcia komputera biurowego, zainstalowanego u użytkownika serwisem technicznym i software'owym;

- Inżynierii systemowej, w zakres której wchodzi: pełna znajomość architektury hardware'u, umożliwiająca zlokalizowanie ewentualnych usterek oraz pełna znajomość software'u, łącznie z umiejętnością samodzielnego tworzenia, modyfikowania i adaptowania programów użytkowych.

Poza tym, na życzenie użytkowników, producent prowadzi szkolenie programistów w zakresie umiejętności formułowania, testowania i uruchamiania programów pisanych w kodzie "komputer biurowy".

Samodzielne tworzenie programów użytkowych przez użytkowników, mimo że w pełni możliwe przy względnej prostocie kodu "komputer biurowy", ze względu na jego dużą pracochłonność i znany deficyt kadr programistów, nie jest przez producenta zalecane. W stosunku do opracowanych programów producent podejmuje się ich atestacji, a zwłaszcza sprawdzenia opłacalności eksploatacyjnej z punktu widzenia wymogów konkretnej konfiguracji.

## 7. Zasady dostawy oprogramowania

Założeniem przyjętym przy upowszechnianiu zastosowań komputera biurowego MERA 302 jest fabryczna produkcja oprogramowania użytkowego i świadczenie pełnego zakresu usług software'owych w zakresie uruchamiania poszczególnych programów u konkretnych użytkowników.

W momencie dostawy sprzętu, jego użytkownik /o ile może wykazać się posiadaniem przeszkolonego operatora/ otrzymuje oprogramowanie podstawowe, w skład którego wchodzi system operacyjny komputera biu-

rowego i zbiór fabrycznych pakietów programów użytkowych. W skład tego zbioru wchodzi obecnie pakiety programów z następujących dziedzin:

- gospodarki materiałowej,
- ewidencji magazynowej,
- księgowości ogólnej i rachuby płac,
- kosztorysowania,
- gospodarki wyrobami gotowymi i fakturowania,
- technicznego przygotowania produkcji.

Biblioteka programów fabrycznych będzie systematycznie rozbudowywana i uzupełniana, w oparciu o analizę dotychczasowych wdrożeń i zgłoszeń potrzeb użytkowników.

Zasadą przyjętą przy tworzeniu programów użytkowych jest wykonywanie ich według szczegółowych założeń opracowanych na podstawie potrzeb konkretnych użytkowników. W okresie wstępnym objęto opracowywaniem najbardziej typowe potrzeby, mogące występować u możliwie największej ilości użytkowników sprzętu.

Każdy program, wchodzący w skład biblioteki fabrycznej programów, został uruchomiony i dokładnie eksploatacyjnie przetestowany u przynajmniej jednego użytkownika. W bieżącym roku gros uruchamianych programów przystosowano do potrzeb przedsiębiorstw przemysłu maszynowego.

Zakład Doświadczalny Oprogramowania przy Instytucie Maszyn Matematycznych, producent oprogramowania komputera biurowego, przygotowany jest do wykonywania adaptacji i modyfikacji istniejących programów użytkowych, przystosowując je zarówno w zakresie postaci danych wejściowych jak i wyjściowych do założeń eksploatacyjnych, jakie otrzymuje od poszczególnych przedsiębiorstw zainteresowanych automatyzacją określonych dziedzin swojej działalności. Podobnie rozbudowa istniejącego zakresu oprogramowania wykonywana będzie przy uwzględnieniu wszystkich wymogów zgłoszonych przez przedsiębiorstwa i instytucje zainteresowane nowymi uruchomieniami w zakresie software'u.

Wszystkie wykonywane obecnie programy przewidują ich eksploatację z wykorzystaniem papierowego nośnika informacji. Charakter nośnika określa w dużym stopniu zakres zastosowań, zwłaszcza wielkość zbiorów danych podstawowych, jakie mogą być użyte do jednoczesnego przetwarzania w reżimie półautomatycznym. Programy, których eksploatacja przewiduje wykorzystywanie dużych zbiorów informacji podstawowej, będą mogły być w prosty sposób dostosowane do eksploatacji na konfiguracjach sprzętowych, rozwiniętych o zewnętrzne urządzenia pamięciowe na nośniku magnetycznym.



Pośród istniejących pakietów programów użytkowych warto przykładowo opisać bliżej następujące zastosowania komputera biurowego:

1/ Zakładanie taryfikatora cen katalogowych robocizny, materiałów, sprzętu i urządzeń oraz sporządzanie na ich podstawie kosztorysów.

Zakładanie taryfikatora /cennika/ na kartach brzeźnie perforowanych ma na celu ułatwienie i zmechanizowanie prac związanych z kosztorysowaniem. Uniwersalny taryfikator cen, robocizny, materiałów, sprzętu i urządzeń ma służyć jako kartoteka pozycji względnie stałych /aktualizowanych raz na 2 - 3 lata w wypadku zmiany cen/.

Sporządzenie kosztorysu polega na:

- wybraniu z kartoteki wszystkich kart obrzeźnie perforowanych uczestniczących w danym kosztorysie,
- podaniu z klawiatury symbolu identyfikacyjnego danej pozycji katalogowej,
- podłożeniu karty pod czytnik,
- automatycznej kontroli zgodności symbolu identyfikującego, podanego z klawiatury, z symbolem na karcie,
- w wypadku niezgodności symboli - zignorowaniu przez maszynę treści karty i podłożeniu pod czytnik karty następnej,
- podaniu z klawiatury ilości,
- automatycznym obliczeniu wartości: robocizny, materiałów, sprzętu i urządzeń oraz wypisaniu wyników w odpowiednich kolumnach. Zakończenie każdego rozdziału kosztorysu polega na automatycznym podsumowaniu wartości w kolumnach, obliczeniu narzutów i dodatków do odpowiednich pozycji oraz wypisaniu sum ogólnych.

2/ Zakładanie kartoteki indeksu materiałowego.

Kartotekę indeksu materiałowego proponuje się prowadzić na kartach brzeźnie perforowanych. Dla każdego rodzaju materiału prowadzona jest oddzielna karta. Pierwszą kartą zbioru będzie karta zawierająca metrykę. Postać karty indeksu materiałowego będzie następująca:

- Pozycja 1 zawiera symbol indeksu materiałowego w postaci alfanumerycznej o długości do 10 znaków;
- Pozycja 2 zawiera nazwę materiału w postaci alfanumerycznej od długości do 55 znaków,
- Pozycja 3 zawiera jednostkę miary w postaci alfabetycznej o długości do 3 znaków,
- Pozycja 4 zawiera cenę materiału w postaci numerycznej o długości do 7 cyfr,
- Pozycja 5 zawiera kod sprawozdawczy w postaci numerycznej o długości do 4 cyfr.

Dane wejściowe wprowadzane są z klawiatury maszyny do pisania, a następnie dziurkowane na kartach brzeźnie perforowanych. Wyperforowane karty należy układać według wzra-

stającej wartości symbolu indeksu materiałowego i przechowywać w odpowiednich pojemnikach.

Po założeniu kartoteki indeksu materiałowego możliwe jest emitowanie indeksu na tabulogram.

Karty brzeźnie perforowane odczytywane są przez czytnik, a następnie wypisywane w żądanej postaci tabulogramu. Okresowo należy przeprowadzać aktualizację indeksu materiałowego, zastępując karty zdezaktualizowane kartami nowo założonymi.

3/ Sprawozdawczość materiałowa, która obejmuje:

- sprawozdania ilościowe /odpowiadające formularzowi GUS GM-1/, dotyczące ważniejszych surowców i materiałów według Systematycznego Wykazu Wyróbów,
- sprawozdania wartościowe /odpowiadające formularzowi GUS GM-11/ z zakresu wartości zapasów i zużycia materiałów.

Dane wejściowe potrzebne do sporządzenia tabulogramu /w takim układzie jak sprawozdanie GUS/ pobierane są z kartoteki ilościowo-wartościowej, sporządzanej okresowo na podstawie dokumentów źródłowych /jak PZ, WZ, RW i inne/.

Danymi wejściowymi są następujące informacje:

- symbol i nazwa materiału /surowca/,
- jednostka miary,
- stan na początek roku,
- wielkość przychodu,
- wielkość rozchodu z rozbiciem na zużycie.

Informacje te zawarte są na kartach brzeźnie perforowanych, stanowiących kartotekę ilościowo-wartościową materiałów.

Przetwarzanie odbywa się w następujący sposób: z klawiatury zostaje wprowadzony do pamięci odpowiedni kod sprawozdawczy dotyczący formularza GUS, który ma być sporządzony. Następnie karty stanowiące kartotekę ilościowo-wartościową są kolejno wczytywane. W przypadku niezgodności kodu sprawozdawczego zawartość karty zostaje zignorowana. W rezultacie brane będą pod uwagę zawartości tych kart, których kod sprawozdawczy podany z klawiatury jest zgodny z kodem karty wczytanej. Po wczytaniu wszystkich kart potrzebnych do sporządzenia sprawozdania następuje wydruk tabulogramu. Wydruk ten jest sterowany odpowiednim programem zawierającym standardowy program drukowania części opisowej formularza.

W przypadku konieczności doliczania niektórych wartości /np przy sporządzaniu sprawozdania typu GM-11 - obliczanie zapasów zbędnych i nadmiernych/, czynności te wykonywane są automatycznie, podobnie jak wydruk tabulogramu.



## OPROGRAMOWANIE KOMPUTERÓW W UKŁADACH AUTOMATYKI I POMIARÓW

### 1. Wstęp

W Przemysłowym Instytucie Automatyki i Pomiarów MERA-PIAP, w ramach prac podstawowych nad oprogramowaniem komputerów na potrzeby automatyki kompleksowej, prowadzonych w ramach problemu węzłowego 06.1.2 oraz w ramach prac aplikacyjnych nad oprogramowaniem komputerów produkowanych na potrzeby automatyki i pomiarów w Zjednoczeniu MERA, prac prowadzonych w ramach problemu węzłowego 06.4.1 i na zlecenia bezpośrednie, sformułowano [1] nomenklaturę i podział oprogramowania służącego tym celom, w sposób niżej podany.

Całość oprogramowania na potrzeby automatyki i pomiarów dzielimy z punktu widzenia zastosowania, rodzaju i pochodzenia.

#### A. Zastosowania oprogramowania

- A1. Rejestracja danych pomiarowych
- A2. Przetwarzanie danych pomiarowych
- A3. Kontrola
- A4. Sterowanie w układach sekwencyjnych i decyzyjnych
- A5. Bezpośrednie sterowanie cyfrowe /ang. DDC/
- A6. Sterowanie nadrzędne - w skali obiektu /lub węzła technologicznego/
  - A6.1. Optymalizacja
  - A6.2. Rozruchy - odstawiania
  - A6.3. Działanie na wypadek awarii
- A7. Zarządzanie przedsiębiorstwem /m. in. wielozakładowym/
  - A7.1. Dokumentacja działania
  - A7.2. Operatywne sterowanie produkcją
  - A7.3. Sterowanie magazynem
  - A7.4. Obliczanie wskaźników działalności przedsiębiorstwa

A8. Zapisywanie modeli procesów sterowanych.

#### B. Rodzaje oprogramowania

- B1. Systemy operacyjne czasu rzeczywistego
- B2. Podprogramy obsługi urządzeń zewnętrznych, m. in. sprzęgających z obiektem
- B3. Oprogramowanie użytkowe
- B4. Uniwersalne pakiety programów
- B5. Programy tłumaczące
- B6. Programy testujące
- B7. Programy diagnostyczne.

#### C. Pochodzenie oprogramowania

- C1. Oprogramowanie operacyjne
- C2. Oprogramowanie podstawowe
- C3. Oprogramowanie użytkowe.

Aktualnie reprezentowany jest pogląd, że producenci jednostek centralnych komputerów i minikomputerów winni dostarczać oprogramowanie operacyjne, a w szczególności - systemy operacyjne czasu rzeczywistego i bootstrap'owe programy testujące jednostki centralnej, kanałów i konwencjonalnych urządzeń zewnętrznych. W ramach Krajowego Systemu Automatyki i Pomiarów POLMATIK winno być przygotowane oprogramowanie podstawowe, a w szczególności podprogramy obsługi urządzeń sprzęgających komputery z kontrolowanymi lub sterowanymi obiektami, uniwersalne pakiety programów dla najczęściej spotykanych zastosowań, programy tłumaczące dla języków problemowych na potrzeby automatyki i pomiarów oraz programy diagnostyczne dla zestawów komputerowych sprzężonych z obiektami i pracujących w trybie czasu rzeczywistego. Natomiast oprogramowanie użytkowe winno być opracowywane na zasadach produkcji wyrobów przez



wyspecjalizowane jednostki gospodarcze i dostarczane odbiorcom w ramach generalnych dostaw układów automatyki kompleksowej lub zautomatyzowanych stanowisk pomiarowych /patrz roz. 5/.

Niniejszy artykuł stanowi próbę przeglądu prac aktualnie prowadzonych w Zjednoczeniu MERA w wymienionych kierunkach, dotyczących oprogramowania komputerów produkowanych w tym Zjednoczeniu.

## 2. Oprogramowanie operacyjne

Podstawowym zestawem sprzętu wchodzącym do produkcji w Zjednoczeniu MERA, a przeznaczonym do realizacji zadań automatyki kompleksowej jest System Modułowy Automatyzacji /SMA/ z komputerami Odra-1325.

Dla zestawów SMA - Odra-1325 opracowano program dyrygent EX2P, stanowiący dostosowaną do potrzeb SMA modyfikację znanych dyrygentów firmy ICL nazywanych egzekutorami /ang. executive/. Egzekutor ten zajmujący ok. 6 k słów 24 bitowych /jest generowany automatycznie dla określonej konfiguracji sprzętu/ realizuje wszystkie funkcje egzekutorów serii Odra-1300 jak np.: realizacja programowa ekstrakodów, prowadzenie pracy dwuprogramowej /każdy program z 3 podprogramami/, obsługa urządzeń wejście-wyjście i pamięci pomocniczych, komunikacja z operatorem za pośrednictwem monitora komputera. Ponadto EX2P realizuje dodatkowe funkcje wynikające ze specyfiki SMA, a w szczególności: przesyłanie informacji pod dowolny adres na magistrali SIAL, pobieranie informacji spod dowolnego adresu oraz przyjmowanie i obsługiwanie przerw generowanych w blokach SMA. Przy tym mechanizm obsługiwanie bloków SMA nie różni się w sposób istotny od mechanizmu obsługiwanie urządzeń zewnętrznych w systemie ICL 1900/Odra-1300/, poza tym, że do/z SMA informacje są transmitowane równolegle w słowach 16 bitowych, a nie w znakach 6-bitowych jak w ICL 1900/Odra-1300/.

Z faktu powstania EX2P jako adaptacji wypróbowanego egzekutora firmy ICL wynikają zarówno pozytywne, jak i negatywne konsekwencje. Do pozytywnych zaliczyć należy: gwarancję realizowalności bogatego oprogramowania ICL 1900/Odra-1300/na komputerach pracujących pod kontrolą EX2P oraz możliwość podłączenia pod kontrolą tego dyrygenta różnorodnych urządzeń wejścia-wyjścia i pamięci pomocniczych produkcji krajowej i angielskiej. Do konsekwencji negatywnych zaliczyć należy mniejszą optymalność działania zestawu SMA - Odra-1325 pod kontrolą EX2P, niż byłoby to możliwe przy opracowaniu od podstaw specjalnego systemu operacyjnego. Dotyczy to szybkości działania i zajętości pamięci operacyjnej.

Dla zestawów SMA - Odra-1325 opracowano również zestaw testów boot-strap'owych, umożliwiający badanie poprawności komunikacji Odry 1325 z każdym z opracowanych bloków SMA.

Drugim zestawem sprzętu opracowywanego w Zjednoczeniu MERA, a przeznaczonego do układów centralnej rejestracji danych /CRD/, centralnej rejestracji i przetwarzania danych /CRPD/, obsługi zautomatyzowanych stanowisk kontrolno-pomiarowych, zautomatyzowanych magazynów i sterowania sekwencyjnego bez optymalizacji - jest system PI, aktualnie kompletowany z minikomputerem Momik 8b wchodzącym w skład automatu obrachunkowego MERA-300.

Dla zestawów PI - MERA-300 nie opracowano jeszcze systemu operacyjnego. Pierwsze opracowane programy użytkowe są w swojej istocie programami boot-strap'owymi, tzn. działają na komputerze pozbawionym systemu operacyjnego. Z uwagi na niewielkie rozmiary pamięci operacyjnej MERA-300 nie przewiduje się opracowania uniwersalnego systemu operacyjnego. Kierunkiem działania będzie opracowanie bloków systemu operacyjnego dla obsługi poszczególnych urządzeń zewnętrznych /czytnik taśmy, dziurkarka taśmy, drukarka mozaikowa, klawiatura alfanumeryczna/, pamięci pomocniczej /pamięć dyskowa/ i przede wszystkim bloku sprzęgającego /BS/ systemu PI oraz innych bloków systemu operacyjnego jak np. zarządzającego pracą wieloprogramową. Z tych bloków, odpowiednio uzupełnionych, będzie się konstruować wyspecjalizowane systemy operacyjne dla określonych klas zastosowań. W ten sposób w roku 1974 będą skonstruowane dwa systemy operacyjne: dla układu CRPD dla cukrowni i dla układu sterowania pojedynczym wielozadaniowym stanowiskiem pomiarowym.

Oczywiście, dla zestawów PI - MERA-300, opracowano również zestaw testów boot-strap'owych, umożliwiających badanie poprawności komunikacji MERA-300 z każdym z opracowanych pakietów systemu PI.

## 3. Oprogramowanie podstawowe

Oprogramowanie podstawowe na potrzeby automatyki i pomiarów dla komputerów produkowanych w Zjednoczeniu MERA jest w tej chwili daleko zaawansowane, ale niestety mało zróżnicowane.

Po szczegółowej analizie wykonanej w 1971 roku w MERA-PIAP i uzgodnieniu z zainteresowanymi uznano, że dla programowania większości zadań realizowanych przez komputery pracujące w układach kompleksowej automatyki wolnozmiennych ciągłych procesów technologicznych, najbardziej odpowiedni jest blankietowy system programowania typu języ-



ka BICEPS firmy General Electric, stosowane w komputerach GE-PAC. W chwili wyboru przez nas tego języka, na świecie pracowało już 450 komputerów zaprogramowanych za pomocą BICEPS-a. W oparciu o BICEPS-a sformułowano blankietowy system programowania SZPAK<sup>x</sup> przystosowany do potrzeb krajowych [2].

System programowania SZPAK wchodzi w skład oprogramowania systemu POLMATIK, nie jest zatem przeznaczony dla określonych komputerów, ale zostaną w niego wyposażone wszystkie komputery i minikomputery przeznaczone do automatyki i pomiarów, które będą produkowane w Zjednoczeniu MERA. System programowania SZPAK umożliwia wygodne programowanie komputerów realizujących zadania:

- centralnej rejestracji danych /CRD/,
- centralnej rejestracji i przetwarzania danych /CRPD/,
- sterowania nadrzędnego, tzn. zmieniania przez komputer wartości zadanych dla regulatorów sprzętowych realizujących zadania stabilizacji,
- regulacji kaskadowej,
- koordynacji międzywęzłowej,
- optymalizacji.

Bezpośrednie sterowanie cyfrowe /ang.DDC/ może być realizowane przez oddzielny pakiet programów, współdziałający ze SZPAK-iem. Zadania, których algorytmy obliczeniowe nie mieszczą się w bibliotece SZPAK-a /ma to miejsce zwłaszcza przy programowaniu zadań optymalizacji/, mogą być programowane w określonym podzbiórze języka Fortran, nazywanym językiem PF [3].

Jednym z istotnych założeń systemu SZPAK jest przyjęcie, że część programów oraz część danych do obliczeń nie mieści się w pamięci operacyjnej komputera, ale musi być w czasie rzeczywistym przemieszczona pomiędzy pamięcią operacyjną i pomocniczą /bębnową lub dyskową/. Programy przemieszczania mieszczą się w systemie SZPAK, w związku z czym organizacja tego kłopotliwego przedsięwzięcia nie obciąża w ogóle programisty, ale jest realizowana automatycznie.

Ponieważ aktualnie Odra-1325 jest jedynym komputerem produkcji Zjednoczenia MERA mogącym realizować wszystkie funkcje, do których przewidziano SZPAK-a, SZPAK jest implementowany w pierwszej kolejności dla zestawów SMA - Odra-1325. System SZPAK będzie dostępny dla użytkowników zestawów w końcu 1974 roku; pierwszym dużym układem automatyki kompleksowej zaprogramowanym za pomocą SZPAK-a, będzie centralny układ sterowania i przetwarzania danych z zastosowaniem komputera w Janikowskich Zakładach Sodowych.

x) SZPAK - skrót od System Zintegrowanego Programowania dla Automatyki Kompleksowej"

W tym miejscu należy zwrócić uwagę na pewne osobliwości implementacji SZPAK-a dla zestawów SMA - Odra-1325. Egzekutor EX2P jest dyrygentem dwuprogramowym w sensie przyjętym w przetwarzaniu danych. To znaczy, że każdy z tych dwóch programów działa na oddzielnych obszarach pamięci, z oddzielnymi urządzeniami zewnętrznymi. Egzekutor EX2P wyklucza jakkolwiek możliwość współdziałania tych dwóch programów, na przykład nie jest możliwe jednoczesne korzystanie przez oba programy z tych samych danych czerpanych z obiektu za pomocą bloków SMA. Dzięki systemowi SZPAK nie jest to, na szczęście, żadna niedogodność. System operacyjny SZPAK-a jest jednym z dwóch programów pracujących pod kontrolą EX2P. Wszystkie, chociażby bardzo liczne programy napisane przez użytkownika, a służące celom przetwarzania danych z obiektu, wyliczaniu sygnałów sterujących itp., stanowią w istocie swojej segmenty programów w języku Fortran. Po translacji i specjalnej konsolidacji z systemem operacyjnym SZPAK-a, wszystkie te programy pracują pod kontrolą SZPAK-a, zwykle rezydując w pamięciach pomocniczych.

Natomiast drugi z programów, które mogą pracować pod kontrolą EX2P może być wykorzystany do celów nie związanych bezpośrednio z obsługiwaniem obiektu w czasie rzeczywistym. Może to być na przykład translator Fortranu, wykorzystywany do tłumaczenia nowych programów PF przeznaczonych do wprowadzenia do systemu. Oczywiście, korzystanie z tego drugiego programu jest możliwe wyłącznie w przypadku pozostania wolnego czasu i miejsca w pamięci operacyjnej komputera Odra-1325.

Dla małych układów CRD i CRPD, dla których są przeznaczone zestawy PI - MERA-300, przewiduje się uproszczoną wersję SZPAK-a. Uproszczenia te będą w większej mierze dotyczyły sposobu wewnętrznej realizacji systemu operacyjnego, translatorów i komolidatorów SZPAK-a, niż sposobu zapisywania programów źródłowych przez programistę. Istotne dla użytkownika będą następujące dwa ograniczenia: brak możliwości tłumaczenia, uruchamiania i zmieniania fragmentów programów w trakcie ich realizacji /co jest możliwe w SZPAK-u/ i konieczność pisania programów uzupełniających w języku MOTIS, będącym językiem symbolicznym minikomputera MERA-300, zamiast używania Fortranu.

Poza SZPAK-iem i jego uproszczoną wersją, system POLMATIK przewiduje co najmniej opracowanie następujących rodzajów oprogramowania podstawowego:

- oprogramowanie diagnostyki systemowej,
- język problemowy na potrzeby sterowania sekwencyjnego,
- język programowania stanowisk pomiarowych i kontrolnych,



- język problemowy na potrzeby sterowania i obsługi magazynów,
- uniwersalny pakiet programów sterowania blokiem energetycznym i inne, w miarę potrzeb.

Szczegółowe wymagania i założenia dla wyżej wymienionych jednostek oprogramowania podstawowego nie zostały jeszcze sformułowane.

#### 4. Oprogramowanie użytkowe

Dążeniem Zjednoczenia MERA jako producenta zestawów sprzętowych dla potrzeb automatyki i pomiarów, /aktualnie są nimi: SMA - Odra-1325 i PI - MERA-300/, jest zapewnienie, aby możliwie duży procent oprogramowania użytkowego mógł powstawać przy wykorzystaniu oprogramowania podstawowego. Ponieważ jednak odpowiednie zestawy sprzętowe zostały udostępnione zespołom programistów dopiero w ostatnim czasie, a jednocześnie wielki nacisk potrzeb spowodował, że jest przewidywane powielanie i przekazywanie w/w zestawów użytkownikom bez czekania na zakończenie prac nad oprogramowaniem podstawowym - stało się rzeczą nieuchronną, że w latach 1974-75 wiele programów użytkowych musi być napisanych bez użycia bardziej zaawansowanych pomocy. Na dzień dzisiejszy możliwości pisania programów użytkowych są następujące:

Dla zestawów SMA - Odra-1325 można pisać programy w języku PLAN, który jest językiem symbolicznym komputerów ICL-1900/Odra-1300/. Jest to wysoko rozwinięty język symboliczny z możliwościami symbolicznego adresowania, segmentowania, definiowania i wykorzystywania makroinstrukcji oraz z biblioteką podprogramów i funkcji pomocniczych /m. in. matematycznych, przetwarzaniowych i obsługi pamięci pomocniczych/. Na poziomie instrukcji język ten przewiduje komunikowanie się z urządzeniami zewnętrznymi za pomocą instrukcji PERI z odpowiednimi obszarami kontrolnymi określającymi zadania związane ze specyfiką urządzenia zewnętrznego, którym może być blok sprzęgający SMA. Zatem program napisany w języku PLAN, a realizujący komunikację z SMA za pomocą instrukcji PERI, może być tłumaczony za pomocą standardowych translatorów języka PLAN dostępnych w bibliotece komputera Odra-1325, a następnie może pracować pod kontrolą egzekutora EX2P.

Dla zestawów PI - Mera-300 istnieje możliwość pisania programów w języku symbolicznym MOTIS z indywidualnym dołączaniem opracowanych już podprogramów obsługujących poszczególne operacje bloku sprzęgającego BS systemu PI.

Na zakończenie tego rozdziału należy zaznaczyć, że dotyczył on oprogramowania zestawów przewidywanych do rozpowszechniania przez pionierów automatyki i pomiarów Zjednoczenia MERA. Obok tego, w pionie informatyki Zjednoczenia MERA są prowadzone prace indywidualne nad określonymi zastosowaniami komputerów w automatyce i pomiarach. Prace te powstają w Instytucie Maszyn Matematycznych w Warszawie i w Oddziale Śląskim tego Instytutu. Wymienić tu można np. układ sekwencyjny zastosowany przy sterowaniu produkcją poli-propylenu w Płocku. Układy te są wyposażone oczywiście w programy użytkowe.

Programy te są przygotowywane jednostkowo.

#### 5. Tryb opracowywania oprogramowania użytkowego

Oprócz przygotowywania programów indywidualnie przez użytkowników lub wyspecjalizowane przedsiębiorstwa pracujące na ich zlecenie, przewiduje się w Zjednoczeniu MERA przyjęcie określonego postępowania i form organizacyjnych, które zapewniłyby w ramach generalnych dostaw, dostarczenie nabywcom układów automatyki kompleksowej i skomputeryzowanych stanowisk kontrolno-pomiarowych kompletnego oprogramowania użytkowego potrzebnego do realizowania przez sprzęt wymaganych przez nabywcę celów.

Sugerowany przez Ośrodek Automatykacji Kompleksowej i Systemów Cyfrowych MERA-PIAP, ale jeszcze nie zatwierdzony, tryb postępowania przy opracowywaniu oprogramowania użytkowego byłby następujący:

1/ Użytkownik /sam lub z pomocą Pracowni Projektowej albo placówki naukowo-badawczej/ formułuje wymagania na układ automatyki kompleksowej lub skomputeryzowane stanowisko kontrolno-pomiarowe.

2/ Wymagania te są podstawą dla zlecenia opracowania Założeń Techniczno-Ekonomicznych, które powstają równolegle w dwóch aspektach: sprzętu i oprogramowania. Dla procesów technologicznych pilotujących Założenia te opracowują jednostki zaplecza naukowo-badawczego /Instytut, OBR, Zakład Doświadczalny/, dla procesów technologicznych powtarzanych - Pracownie Projektowe.

Prace w zakresie pp. 1 i 2 wyceniane są na podstawie pracochłonności.

3/ Na podstawie zatwierdzonych Założeń na oprogramowanie, Generalny Dostawca zleca wyspecjalizowanemu przedsiębiorstwu wyprodukowanie oprogramowania. Powstałe w ten sposób oprogramowanie stanowi wyrób, wyceniony wg zasad cennikowych [4]. W skład dostawy wchodzi m. in. uruchomienie oprogramowania w warunkach laboratoryjnych. W tym celu



albo producent oprogramowania posiada własne zestawy sprzętu i symulatory obiektu albo w umowie zastrzega konieczność wyprzedzeniowej kompletacji sprzętu przez przedsiębiorstwo kompletujące i oddanie go do eksploatacji laboratoryjnej w celu umożliwienia uruchomienia oprogramowania.

4/ Jeżeli uruchomienie oprogramowania nastąpiło na sprzęcie własnym producenta oprogramowania, wtedy oprogramowanie to musi przejść jeszcze jeden test w warunkach laboratoryjnych na docelowym zestawie sprzętu, co jest łączone z testowaniem zestawu sprzętu cyfrowego i komputera dokonywanym przez kompletatora, przed dostawą na obiekt. Uruchamianie oprogramowania na sprzęcie własnym producenta oprogramowania daje tę korzyść, że nie jest potrzebne duże wyprzedzenia kompletacji dostaw. Ale dobrze jest, jeżeli uruchamianie oprogramowania i testowanie sprzętu docelowego odbywają się w tym samym fizycznie laboratorium, a to w celu umożliwienia korzystania z tych samych urządzeń symulujących obiekt, które są bardzo kosztowne, a do wykorzystania docelowego na obiekcie nieprzydatne.

5/ Kolejną fazę realizacji dostawy układu automatyki kompleksowej lub skomputeryzowanego stanowiska kontrolno-pomiarowego jest rozruch na obiekcie. Dotyczy to zarówno sprzętu, jak i oprogramowania. Tak jak ostateczna regulacja sprzętu jest możliwa tylko w realnych warunkach na obiekcie, tak i ostateczne "podstrojenie" oprogramowania jest możliwe tylko w warunkach rzeczywistych. Właściwe wykonanie poprzednich etapów ma zapewnić minimalizację prac koniecznych do wykonania na obiekcie. Do prac, których wyeliminować nigdy się nie da, należy ustalenie wartości liczbowych współczynników oraz dopasowanie oprogramowania do aktualnych wymagań w czasie rzeczywistym i występujących obciążeń magistrali sprzężenia i jednostki centralnej zestawu komputera. Szczególnie liczne prace wiążą się z ustaleniem optymalnych algorytmów sterowania optymalnego - te ostatnie prace mogą być bardzo długotrwałe i przebiegać równoległe z eksploatacją obiektu. Dlatego prac tych nie można wyceniać na zasadach cennikowych. Jest natomiast oczywiste, że rozruch oprogramowania na obiekcie musi być zlecony producentowi oprogramowania.

6/ Po uruchomieniu oprogramowania i przekazaniu go użytkownikowi, musi ono podlegać gwarancji jak każdy wyrób. W okresie gwarancyjnym producent byłby zobowiązany na

ogólnych warunkach dostaw do usuwania wszelkich stwierdzonych usterek, tj. niezgodności działania oprogramowania w stosunku do Założeń Techniczno-Ekonomicznych na oprogramowanie.

Nie wydaje się natomiast celowe świadczenie usługi zwanej konserwacją oprogramowania. Pod pojęciem konserwacji oprogramowania rozumie się stałe ulepszanie i wzbogacanie oprogramowania oraz poprawianie usterek, które mogą dać skutki w nowych zastosowaniach. Konserwacja oprogramowania jest wymagana w stosunku do oprogramowania podstawowego komputerów pracujących w zastosowaniach informatycznych.

Takie poprawianie oprogramowania użytkowego, pracującego na obiekcie nie jest potrzebne, jeżeli oprogramowanie spełnia postawione założenia, czyli jest wolne od błędów. Wnoszenie zmian do oprogramowania wymaga bowiem posiadania przez użytkownika fachowców programistów, a czasami może przynieść bardzo niepożądane skutki spowodowane efektami ubocznymi lub niedopasowaniem do szczególnych właściwości sprzętu /np. dokonanych przeróbek/, nieznanymi zalecającemu usprawnianie oprogramowania. Należy zatem przyjąć za obowiązującą zasadę, że ulepszenia oprogramowania użytkowego winny być przeprowadzane rzadko, najlepiej w ramach modernizacji obiektu.

Jesteśmy przekonani, że przedstawiony tryb opracowywania oprogramowania użytkowego dla komputerów stosowanych w automatyce i pomiarach zaspokoi w sposób właściwy potrzeby zarówno odbiorców jak i producenta, tj. Zjednoczenia MERA.

#### Literatura

- [1] Orłowski H.: Oprogramowanie komputerów w układach sterowania, "Biuletyn MERA-PIAP", 1971, nr 4/30, s. 59-67.
- [2] Aderek A.: Blankietowy system programowania SZPAK, "Biuletyn MERA-PIAP", 1972, nr 5/37, str. 3-33.
- [3] Orłowski H.: Możliwości wykorzystania języka Fortran do uzupełnienia blankietowego systemu programowania SZPAK, "Biuletyn MERA-PIAP", 1972, Nr 5/37, s. 35-48.
- [4] Białasiewicz J., Orłowski H.: Metodyka określania ceny oprogramowania komputerowych systemów centralnej rejestracji danych i sterowania, "Biuletyn MERA-PIAP", 1973, nr 6 /w druku/.



SPIS ARTYKUŁÓW OPUBLIKOWANYCH W BIULETYNIE "MERA"

W ROKU 1973

Technika

	nr
1. BŁASZKÓW S. - Wrocławskie Zakłady Elektroniczne MERA-ELWRO .....	5
2. CHLEBOWSKI R. - Aparatura kontrolno-pomiarowa dla przemysłu okrętowego produkowana w MERA-KFAP, .....	10
3. CHRZANOWSKI A., JAWORSKI M. - Produkcja kół zębatych z poliacetalu w MERA-PAFAL .....	3
4. DUSZYŃSKI S. - Przemysłowy Instytut Automatyki i Pomiarów MERA-PIAP w Warszawie .....	5
5. FRANKOWSKI J., GOSTYŃSKI W. - Przedsiębiorstwo Kompleksowej Automaty-zacji MERAMONT .....	5
6. GOSIEWSKI K., KUBIT S. - Problemy stabilności magnesów trwałych stosowa-nych w przetwornikach pomiarowych .....	2
7. GRABOWSKI J., OŚRODEK W., WOJNA J. - Zakłady Mechanizmów Precyzyj-nych MERA-PREZAM .....	5
8. GRENIEWSKI M. J. - Wytwarzanie oprogramowania metodami przemysłowymi	12
9. HAWRYLUK J. - Laboratorium Cbliczeń i Modelowania MERA-PIAP .....	9
10. JAGOSZEWSKI M. - Wrocławskie Przedsiębiorstwo Pomiarów i Automatyki Elektronicznej MERA-ELMAT .....	5
11. JANICKI W. - Kujawska Fabryka Monometrów MERA-KFM .....	5
12. JĘDRZEJKO H. - Warszawskie Zakłady Urządzeń Informatyki MERAMAT .....	5
13. JUNG T. - Wybrane zagadnienia z technologii produkcji liczników energii elektrycznej .....	3
14. Zjednoczone Zakłady Elektronicznej Aparatury Pomiarowej MERA-ELPO .....	5
15. KAMBURELIS T. - Elektroniczna Maszyna Cyfrowa R30 Jednolitego Systemu RIAD .....	1
16. KASPERKIEWICZ A. - Automatykacja wytwórni kwasu siarkowego w oparciu o system automatyki pneumatycznej PNEFAL .....	8
17. KRĘŻAŁEK Z. - Zakłady Automatyki MERA-POLNA .....	5
18. KNAP E., SASIN M. - Zakłady Wytwórcze Aparatury Precyzyjnej MERA-PAFAL .....	5
19. KONON E. - Tendencje rozwojowe produkcji czujników do pomiaru temperatury w MERA-KFAP .....	10
20. KOT R. - Zagadnienie zdalnego sterowania odbiornikami energii elektrycznej za pomocą sygnałów o częstotliwości akustycznej .....	3
21. KOWALSKI L. - Perspektywy zastosowania optoelektroniki w aparaturze kontrol-no-pomiarowej i środkach automatyzacji /cz. I/ .....	11
22. KOWALSKI L. - Perspektywy zastosowania optoelektroniki w aparaturze pomia-rowej i środkach automatyzacji /cz. II/ .....	11
23. KOWALSKI L. - Wybrane problemy rozwoju środków automatyzacji /cz. I/ .....	6
24. KOWALSKI L. - Wybrane problemy rozwoju środków automatyzacji /cz. II/ Ele-menty automatyki hydraulicznej .....	9
25. KOŹMINSKI W. - Urządzenie typu UTP-01 do temperowania prętów na automaty	6
26. KRUSZELNICKI B., MROCZEK A. - Pamięć bębnowa PB-304 .....	2
27. KUDŁA M., WCISŁO R. - Regulator współczynnika mocy model RC-4 .....	3



28. KUDŁA M., WICHER Z. - Zagadnienie powtarzalności wskazań liczników w świetle własnych badań .....	3
29. KURC H. - Krakowska Fabryka Aparatów Pomiarowych MERA-KFAP .....	10
30. LASOCKI J. - Automatykacja cementowni .....	3
31. ŁUCKOŚ Z., ROZUMOWSKI J. - Zastosowanie transporterów taśmowych w montażu liniowym w ZWAP MERA-PAFAL .....	3
32. MALENDĄ M., FÓRMANEK J. - Automatykacja bloków energetycznych przez Zakłady Automatyki Przemysłowej MERA-ZAP .....	8
33. MARCINKOWSKI W., NOWAK F., SZACHRAJ A. - Zakłady Automatyki Przemysłowej MERA-ZAP .....	5
34. MĄDRZAK Cz., MATUSZEWSKI B. - Rozwojowe problemy produkcji indukcyjnych liczników energii elektrycznej w aspekcie ich nowoczesności, jakości i asortymentu .....	3
35. MIAŁKOWSKI B. - Wpływ uniwersalnych przyrządów składanych na mechanizację prac w przemyśle maszynowym .....	6
36. MURZYN T., BOCHEŃSKI S., GOŁASKI A., LORYCH A., LUBICZ A. - Automatykacja procesów klimatyzacyjnych w Polsce .....	8
37. OGRODNIK K. - Rozwój Zakładów Wytwórczych Aparatury Precyzyjnej MERA-PAFAL w Świdnicy .....	3
38. OKRASA W., KRZYWIŃSKI K. - Zakłady Mechaniczno-Precyzyjne MERA-BŁONIE .....	5
39. OMYLIŃSKA B. - Regulatory ciśnienia i przepływu bezpośredniego działania typu BRU .....	9
40. ORŁOWSKI H. - Oprogramowanie komputerów w układach automatyki i pomiarów .....	12
41. OTTO R., MAYDELL T. - Najnowsze osiągnięcia w dziedzinie przemysłowych przetworników pomiarowych .....	10
42. PAWLAK Z. - Automatykacja sterowania wirówką paliwa .....	8
43. PIOTROWICZ J. - Zakłady Mechaniki Precyzyjnej MERA-WAG .....	5
44. PORĘBSKI Z. - Minikarty - nowy kierunek APD? .....	1
45. Przedsiębiorstwo Doświadczalne Produkcji Aparatury Kontrolno-Pomiarowej .....	5
46. RATAJSKI J. - Zakłady Wytwórcze Przyrządów Pomiarowych ERA .....	5
47. ROMER J. - Problemy technologii lutowania sprzętu elektronicznego .....	2
48. SADY R. - Wykonywanie łącznika rdzeni na prasie "Hydomat" .....	3
49. SIECZKO J. - Biuro Projektów Urządzeń Informatyki INFOPROJEKT .....	5
50. SENYSZYN J. - Lubuskie Zakłady Aparatów Elektrycznych MERA-LUMEL .....	5
51. SKOCZYŁAS M. - Problemy elektronicznego pomiaru energii elektrycznej .....	3
52. SKRZYŃECKI E. - Układy sygnalizacji z monitorem typu USp-2M .....	11
53. STATECZNY B., WILIMOWSKI J. - Spawanie części i zespołów aparatury kontrolno-pomiarowej w MERA-KFAP .....	10
54. SZADURSKI K. - Automatykacja cukrowni Xanthi .....	8
55. SZOTT T. - Niektóre elementy mechanizacji prac narzędziowych w MERA-KFAP .....	10
56. SZYPOWSKI S., WĘGROWSKI J. - Przedsiębiorstwo Automatyki Przemysłowej MERA-PNEFAL .....	5
57. TOMASIK Z. - Urządzenie do elektromagnetycznego kształtowania metali .....	4
58. TOMASZEWICZ H. - Wybrane zagadnienia doboru układów mostkowych w rejestratorach elektrycznych .....	10
59. WAJCEN M. - Informacja o seminarium na temat minikomputerów "Minicomputer - Forum" .....	9
60. WIDUCH A., MUSIOŁ J. - Blachy stalowe pokrywane tworzywami organicznymi .....	11
61. WILIMOWSKI J. - Technologiczne metody wytwarzania w MERA-KFAP i perspektywy rozwojowe .....	10



62. WILCZEK W. - Układ sygnalizacji optycznej USP-1M wykonany z elementów "Merałog"	
63. WIŚNIEWSKI A. M., PODLEWSKI S. J. - Komputer biurowy MERA 302 - architektura i zasady działania .....	12
64. WIŚNIEWSKI A. M. - Systemy operacyjne JS EMC .....	12
65. WOJCIĘCHOWSKI J. - Metody kontroli i odbioru liczników energii elektrycznej .....	3
66. WYMYSŁOWSKI W. - Przedsiębiorstwo Doświadczalne Produkcji Urządzeń Peryferyjnych	5
67. WYDŹGA S. - Indukcyjne czujniki położenia / inicjatory / w systemie URSAMAT .....	4
68. ZAJĄCZKOWSKI Z. Mierniki przepływomierzy z elementami sprężystymi produkcji MERA-KFAP .....	10
69. ZAKRZEWSKI K. - Układ do dwustopniowej sygnalizacji stanów obiektów .....	11
70. ZIELIŃSKI J., KOZŁOWSKI Z. - Zakłady Aparatury Elektrycznej MERA-REFA .....	5
71. ZIEMBICKI A. - Podsystem Centralnej Rejestracji i Kontroli CRIK w Systemie Modułów Automatykacji /SMA/ .....	1
72. ŻYBURA E. - Laboratoryjne badanie niezawodności przekształtników półprzewodnikowych .....	1

### Ekonomika i Organizacja

73. Aktualna organizacja systemu generalnych dostaw maszyn i urządzeń dla potrzeb inwestycyjnych w CSRS .....	11
74. BIENKOWSKA I. - Aktualny stan ochrony własności przemysłowej i rzecznictwo patentowe w MERA-KFAP .....	10
75. BIM L. Gospodarka magazynowa na przykładzie WZE MERA-ELWRO we Wrocławiu ....	6
76. BIM L. - Kierunki usprawnienia systemu zaopatrzenia w branży informatyki .....	9
77. BIM L. - Krajowa konferencja naukowo-techniczna na temat unifikacji materiałów .....	1
78. BIM L. - Rozliczanie zużycia materiałów galwanicznych w przedsiębiorstwach przemysłowych .....	8
79. BIM L. - Stan gospodarki materiałowej w przedsiębiorstwach Zjednoczenia "Mera" na tle wyników roku 1972 .....	4
80. BIM L. - Znaczenie gospodarki materiałowej w przedsiębiorstwie przemysłowym w systemie Wielkiej Organizacji Gospodarczej .....	7
81. CZARNUL J. - Próba oceny jakości dokumentacji .....	6
82. CZARNUL J. - Uwagi o organizacji przygotowania Systemu Sterowania Jakością .....	1
83. CZERWIŃSKI S. - Zatrudnienie, fundusz płac i system zachęt materialnych w warunkach nowych zasad działalności gospodarczej Zjednoczenia "Mera" .....	7
84. GRABIŃSKI A. - Szkolenie pracowników Centrali Zjednoczenia "Mera" w zakresie informatyki .....	9
85. JACKOWICZ R. - Nowe zasady działalności gospodarczej Zjednoczenia "Mera" /artykuł wstępny/ .....	2
86. J. R. - Zmiany organizacyjne w Zjednoczeniu Przemysłu Automatyki i Aparatury Pomiarowej "Mera" w 1972 .....	1
87. KYCIA H. - System Informacji o Realizacji Inwestycji - SIRI .....	1
88. KURC H. - Rozwój eksportu MERA-KFAP .....	10
89. MERA-METRONEX na XLII Międzynarodowych Targach Poznańskich .....	5
90. MUŁKA Z. - Zasady współdziałania organizacyjno-produkcyjnego MERA-KFAP z Oddziałem Zamiejscowym w Limanowej .....	10
91. PAMUŁA R. - Rozwój ekonomiczny i techniczny Krakowskiej Fabryki Aparatów Pomiarowych MERA-KFAP .....	10
92. PED A. - Aktualny stan wdrażania i plan prac projektowo-wdrożeniowych Systemu SIKOP-MERA/1300 w roku 1973 .....	6
93. PED A. - Aktualny stan wdrażania systemu SIKOP-MERA /1300 i plan prac projektowo-wdrożeniowych w roku 1973 /cz. II/ .....	7



94. PEDA E. - Informacja o aktualnym stanie wdrażania w Centrali Zjednoczenia "Mera" Systemu Informatycznego "Zdolności Produkcyjne i Inwestycje" .....	4
95. PORĘBSKI Z. - Banki danych w systemie zarządzania .....	4
96. PRZYGODA J., PORĘBSKI Z. - Urządzenia organizacyjno-techniczne za granicą i w kraju .....	2
97. SIKORSKI T. - Podstawowe zadania Zjednoczenia "Mera" w zakresie programowania i prognozowania rozwoju .....	11
98. SKRĘTA S. - Elementy systematycznego planowania rozwoju wyrobów .....	2
99. SPRAWSKI R. - Nowe zasady działalności gospodarczej Zjednoczenia Przemysłu Automatyki i Aparatury Pomiarowej "Mera" .....	7
100. SUSZWEDYK R. - Usprawnienie organizacji pracy i poprawa warunków bhp w oddziale montażu wskaźników LZAE "Mera-Lumel" .....	4
101. TRNKA M. - Ocena pracy rzeczoznawców kontroli jakości w Zakładach "Mera" po trzech latach działalności .....	6
102. WAKSBERG J. - Zasady nowego systemu finansowania w Zjednoczeniu "Mera" .....	7
103. WANAT J. - Wynalazczość pracownicza w MERA-KFAP .....	10



Cena 43.- zł

Pren. roczna 516.- zł

