

P O L S K A   A K A D E M I A   N A U K  
ZAKŁAD SYSTEMÓW AUTOMATYKI KOMPLEKSOWEJ

P.3427/77

# PODSTAWY STEROWANIA

KWARTALNIK

Tom 7 — Zeszyt 1

---

GLIWICE—KRAKÓW—WARSZAWA 1977

## TREŚĆ

T. Kamburelis, A. Zasada, Współbieżna emulacja komputerowa — stronicowa . . . . .	3
T. Czachórski, Przybliżone rozwiązania dokładnych modeli i dokładne rozwiązania modeli przybliżonych w ocenie efektywności złożonych układów komputerowych . . . . .	11
J. Czapnik, Algorytm rozdziału zasobów w systemach wieloprogramowych . . .	25
A. Dzierżykraj, Analityczne modelowanie programów o strukturze wielozadaniowej. . . . .	33
R. Jakubowski, A. Szele, Grafy funkcyjne jako rozszerzenie sieci Petriego. II. Topologiczna i algebraiczna interpretacja procesów markowania . . . . .	51
T. Kaczorek, Projektowanie śledzących wielowymiarowych układów liniowych poddanych zdeterminowanym zakłóceniom . . . . .	59
R. Jakubowski, A. Szele, Zastosowanie języków formalnych i rozmytych automatów . . . . .	69

## CONTENTS

T. Kamburelis, A. Zasada, Concurrent Computer Emulation . . . . .	3
T. Czachórski, Approximate Solutions of Exact Models and Exact Solutions of Approximate Models for Computer Systems Performance Evaluation . . .	11
J. Czapnik, An Algorithm for Resource Allocation in Multiprogramming System . . . . .	25
A. Dzierżykraj, Formal Models for Multitask Processing . . . . .	33
R. Jakubowski, A. Szele, Functional Graphs as an Extension of Petri Nets. II. Topological and Algebraical Interpretation of Marking Processes . . . . .	51
T. Kaczorek, Design of Multivariable Linear Tracking Systems with Deterministic Disturbances . . . . .	59
R. Jakubowski, A. Szele, Applications of Formal Languages and Fuzzy Automata in Problem Solving . . . . .	69

## СОДЕРЖАНИЕ

T. Камбурелис, А. Засада, Параллельная компьютерная эмуляция . . . . .	3
T. Чахурски, Приближённые решения точных модели и точные решения приближённых модели для оценки эффективности сложных вычислительных систем . . . . .	11
Я. Чапник, Алгоритм распределения ресурсов в многопрограммированных системах . . . . .	25
А. Держыкрай, Аналитическое моделирование программ с многозадачной структурой . . . . .	33
Р. Якубовски, А. Шелц, Функциональные графы, как расширение сети Петри. II. Топологическая и алгебраическая интерпретация процессов маркировки. . . . .	51
Т. Качорек, Проектирование многомерных линейных следящих систем при детерминистических возмущениях . . . . .	59
Р. Якубовски, А. Шелц, Применение формальных языков и размытых автоматов для решения задач . . . . .	69

P O L S K A   A K A D E M I A   N A U K  
ZAKŁAD SYSTEMÓW AUTOMATYKI KOMPLEKSOWEJ

---

P O D S T A W Y   S T E R O W A N I A

KWARTALNIK



tom 7 — zeszyt 1

GLIWICE — KRAKÓW — WARSZAWA — 1977

---

P A Ń S T W O W E   W Y D A W N I C T W O   N A U K O W E

KOMITET REDAKCYJNY

*Redaktor naczelny:* Stefan Węgrzyn (Gliwice)  
*Zastępcy redaktora naczelnego:* Jerzy Bromirski (Wrocław),  
Czesław Olech (Warszawa)

*Adres Redakcji*

Zakład Systemów Automatyki Kompleksowej PAN  
Gliwice, ul. Zwycięstwa 21, tel. 91-05-85

PAŃSTWOWE WYDAWNICTWO NAUKOWE  
ODDZIAŁ W KRAKOWIE, Kraków, ul. Smoleńsk 14

Nakład 503 + 117 egz. Ark. wyd. 6,00; ark. druk. 5<sup>12</sup>/<sub>16</sub>.  
Papier druk. sat. kl. III, 70 × 100, 80 g. Do składania oddano  
w grudniu 1976 r. Podpisano do druku w marcu 1977 r.  
Druk ukończono w marcu 1977 r.

Zam. 1171/76

C-34

Cena zł 15.—

DRUKARNIA UNIWERSYTETU JAGIELLOŃSKIEGO  
Kraków, ul. Czapskich 4

## Współbieżna emulacja komputerowa-stronicowa

THANASIS KAMBURELIS, ANDRZEJ ZASADA

*(Maszynopis wpłynął 15 września 1976)*

### 1. Wstęp

Szybki rozwój dokonujący się w elektronice oraz w organizacji logicznej komputerów powoduje, że nowo projektowane komputery posiadają odmienne zasady działania od poprzednich komputerów. Zatem oprogramowanie nowych komputerów nie jest zgodne z oprogramowaniem starych komputerów. W związku z tym powstaje problem automatycznego przeniesienia istniejącego programowego dorobku starych komputerów [1], często o wielomilionowej wartości.

W niniejszej pracy rozważa się metodę emulacji wirtualnej (zwaną stronicową), w której zakłada się podział pamięci na jednolicie adresowane bloki zwane stronicami. Trzeba od razu wyjaśnić, że termin emulacja wirtualna nie oznacza, że komputer emulujący musi być wyposażony w konwencjonalną pamięć wirtualną [7] [8]. Rozważana w naszej pracy metoda emulacji wirtualnej dotyczy komputerów ODRA — RIAD, czyli sposobu przenoszenia oprogramowania maszyn serii ODRA 1300 [4] do maszyn JEDNOLITEGO SYSTEMU „RIAD” [5] [6].

W rozważanej stronicowej metodzie emulacji wirtualnej zakłada się, że:

a) zostanie opracowany prosty i niezależnie działający procesor (PE) do wykonywania programów maszyny emulowanej,

b) procesor PE zostanie wyposażony w małą pamięć lokalną PAL (np.  $1 \div 4$  K słów), w której przechowywać się będzie tylko aktywne fragmenty wykonywanych programów (i danych) maszyny emulowanej,

c) procesor PE będzie współpracował z procesorem głównym (PG) za pośrednictwem standardowego interfejsu wejścia-wyjścia maszyny emulującej (tj. za pośrednictwem szybkiego kanału),

d) zostanie opracowany algorytm minimalizujący liczbę transmisji bloków informacji przesyłanych pomiędzy dwoma poziomami pamięci roboczej — tj. głównej pamięci operacyjnej procesora PG i pamięci lokalnej procesora PE — oraz minimalizujący łączny czas wykonania programu emulowanego przy zadanych ograniczeniach (szybkość procesorów PG i PE, szybkość kanału, pojemność PAL, itp.),

e) nie dokona się żadnych zmian zarówno w sprzęcie, jak i w systemie operacyjnym maszyny emulującej. Opracuje się natomiast tylko program Emulator (na poziomie programów problemowych), który spełnia te same funkcje co w innych metodach emulacji. Program Emulator organizuje również tzw. program kanałowy (tj. informacje sterujące przesyłaniem danych przez kanał).

## 2. Metoda stronicowa

### 2.1. Wprowadzenie

W metodzie stronicowej zakłada się, że pamięć operacyjna procesora głównego (PG) jest podzielona na obszary zwane stronicami o jednakowej wielkości (np. po 256 słów maszynowych). Adres stronicy w pamięci operacyjnej (PAO) nazwiemy umownie adresem realnym stronicy, zaś zawartość stronicy określać będziemy jako blok informacji.

Pamięć lokalna (PAL) procesora emulacji (PE) składa się również z pewnej ilości stronic o tej samej wielkości co stronicie pamięci operacyjnej (rys. 1). W rozważaniach emulacji wirtualnej zakłada się oczywiście, że pojemność pamięci PAL jest bardzo mała (np.  $1 \div 4$  K słów) w stosunku do pojemności pamięci PAO (która przeważnie wynosi od 64 do 1024 K słów w EMC Jednolitego Systemu).

Podstawową funkcją małej pamięci lokalnej PAL jest przechowywanie w niej tych bloków programu emulowanego (rezydującego w PAO), z których pochodzą aktualnie wykonywane (w procesorze PE) fragmenty programów i danych. W związku z tym, że w pamięci PAL można na ogół przechowywać nie cały program emulowany, lecz tylko małe jego fragmenty, powstaje problem wyboru algorytmu alokacji stronic pamięci PAL blokom informacji lub wyboru stronicy pamięci PAL, z której trzeba usunąć blok informacji (ewentualnie przechować go w PAO) w celu zwolnienia miejsca dla nowego żadanego bloku informacji. Wybór stronicy, z której usuwa się blok informacji z PAL, musi być dokonywany według rozsądnej strategii w celu zminimalizowania ilości transmisji bloków informacji z PAO do PAL i na odwrót. Idealną strategią byłby oczywiście taki algorytm, który usuwa z pamięci PAL ten blok, do którego odwołania ze strony procesora PE będą miały miejsce później niż do dowolnego innego bloku w pamięci PAL. Niemniej strategia ta wymaga znajomości kolejności wykorzystania bloków informacji w przyszłości przez program emulowany, lecz w praktyce kolejność ta jest nie znana.

Powyższy problem wymiany bloków informacji pomiędzy różnymi pozio-

mami pamięci występuje również w komputerach wyposażonych w konwencjonalną pamięć wirtualną, gdzie stosuje się praktycznie metody opierające się na informacji opisującej dotychczasowy przebieg wykorzystania bloków programu oraz na ekstrapolacji tego przebiegu. Dotychczas zaproponowano różne algorytmy wymiany bloków, których omówienie znajduje się w polskiej publikacji [8].

Stosowane w maszynach wirtualnych algorytmy wymiany bloków informacji są realizowane środkami programowymi (tj. przez systemy operacyjne) i dlatego mogą być oparte o rozbudowane informacje opisujące dotychczasowy przebieg zachowania się programu wykonywanego. W rozważanych natomiast problemach emulacji współbieżnej, zakłada się techniczną (układową lub mikroprogramową) realizację algorytmu wymiany bloków pomiędzy pamięciami PAO i PAL. Dlatego algorytm ten powinien być prostą regułą alokacji stronice pamięci lokalnej.

## 2.2. Algorytm alokacji dynamicznej stronice PAL

Proponowany niżej algorytm dynamicznej alokacji stronice PAL żądanym blokom rozkazów i danych programu emulowanego jest modyfikacją algorytmu znanego pod nazwą Least Recently Used (LRU), w którym usuwa się najdawniej używany blok [8, 9, 10, 11, 12, 13].

Proponowany w tej pracy algorytm zakłada, że pamięć lokalna PAL procesora PE (rys. 1) jest dodatkowo wyposażona w zestaw tzw. Rejestrów Adresów Realnych (RAR). Ilość  $n$  rejestrów RAR równa się ilości stronice pamięci PAL (ponumerowanych od 0 do  $n-1$ ) i każdy z tych rejestrów jest jednoznacznie związany z jedną stronicyą tej pamięci. Rejestr związany z  $i$ -tą stronicyą pamięci PAL, tj. PAL[ $i$ ], oznaczamy będziemy przez RAR[ $i$ ], gdzie  $i = 0, 1, \dots, n-1$ .

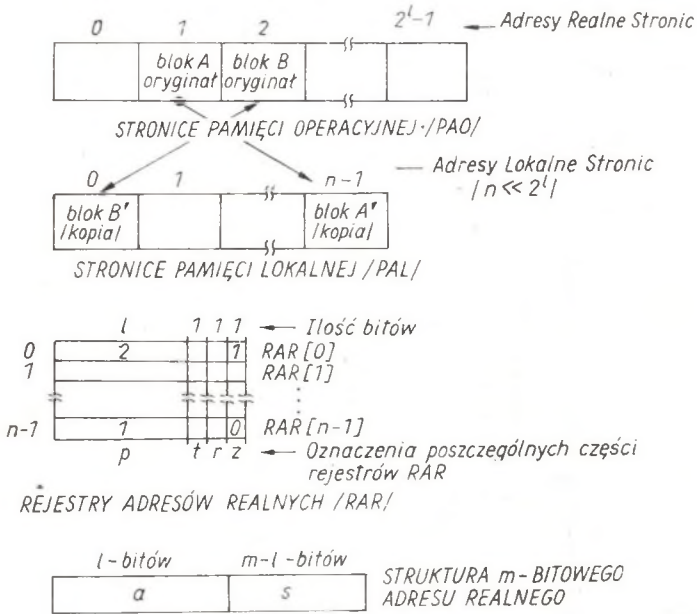
Rejestry RAR mają długość  $l+3$ -bitów, gdzie liczba  $l$  dobierana jest tak, aby można było bezpośrednio adresować wszystkie bloki w PAO (w maszynach JS EMC adresy pamięciowe są 24-bitowe i przy założeniu, że wielkość bloku wynosi 256 słów lub 1024 bajtów, to  $l = 14$  bitów).

W  $l$  bitach rejestru RAR[ $i$ ] pamięta się adres realny bloku informacji zapisanego w stronicy PAL[ $i$ ]. Adres ten służy do identyfikacji bloku w czasie wykonywania programu emulowanego. Natomiast w pozostałych trzech bitach ( $t, r, z$ ) rejestru RAR[ $i$ ] przechowuje się pewne dodatkowe informacje dotyczące aktualnej aktywności bloku zapisanego w obszarze PAL[ $i$ ]. Bit zmian  $z = 0$  wskazuje, że kopia bloku informacji znajdująca się w PAL[ $i$ ] jest zgodna z oryginałem tego bloku znajdującym się w PAO (jeśli zaś  $z = 1$ , to kopia różni się od oryginału). Bit  $z$  jest ustawiany w stanie 1, przy operacjach zapisu (przez procesor PE) do stronicy PAL[ $i$ ], lub w stanie 0 przy ładowaniu nowego bloku do tej stronicy.

Bit referencyjny  $r = 1$  wskazuje, że miało miejsce odwołanie (odczyt

lub zapis) procesora PE do stronicy związanej z rejestrem RAR zawierającym ten bit  $r$ . Bit  $r$  przyjmuje stan 0 przy ładowaniu nowego bloku do odpowiedniej stronicy PAL lub przy braku „świeżych” odwoływań do tej stronicy (patrz rys. 2).

Bit czasu referencji  $t$  został wprowadzony dla rejestracji odwoływań, które miały miejsce w ostatniej kadencji pracy procesora PE, gdzie przez kadencję pracy procesora PE rozumie się okres pracy, jaki upływa pomiędzy dwoma



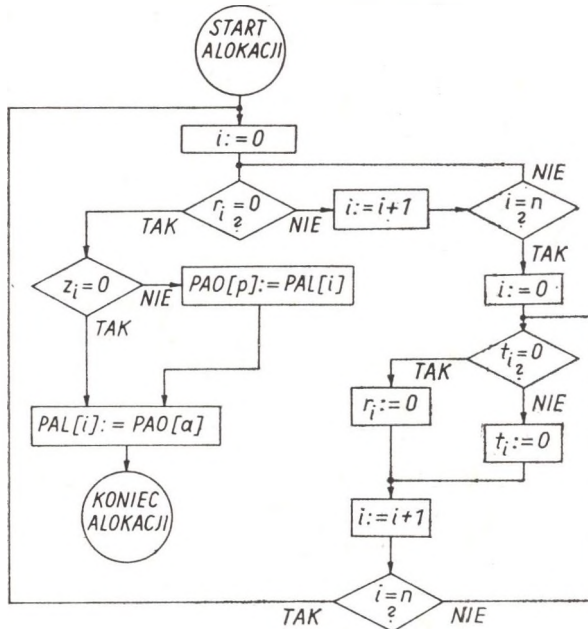
Rys. 1. Struktura pamięci w emulacji wirtualnej opartej o metodę stronicową. Znaczenie części rejestrów RAR:  $z$  — bit zmian ( $z = 1$ , gdy kopia bloku  $\neq$  od jego oryginału);  $r$  — bit referencyjny ( $r = 1$ , po operacjach ZAPIS DO PAL lub ODCZYT Z PAL);  $t$  — bit czasu referencyjnego;  $p$  — pole adresu realnego bloku zapisanego w odpowiedniej stronicy PAL (rejestr RAR  $i$  i związany jest ze stronicą PAL[ $i$ ], gdzie  $i = 0, 1, \dots, n-1$ );  $a$  — adres realny bieżącej stronicy ( $0 < a < 2^l$ );  $s$  — adres słowa wewnątrz danej stronicy

kolejnymi przepelnieniami pamięci PAL (przepelnienie charakteryzuje się brakiem wolnych stronic dla przyjęcia nowego żądanego bloku). Jeśli na końcu danej kadencji pracy bit  $t = 1$ , oznacza to, że w tej kadencji miały miejsce jakieś odwołania do odpowiedniej stronicy. W przeciwnym przypadku  $t$  pozostaje w stanie 0, w którym zostaje wprowadzony na początku danej kadencji pracy.

Szczegółowy algorytm alokacji dynamicznej stronic PAL ilustrowany jest na rys. 2. Procesor PE przechodzi do wykonania algorytmu alokacji, gdy żądany jest blok, którego kopia nie znajduje się w PAL. Algorytm alokacji bada w pierwszym kroku bity referencyjne poszczególnych rejestrów RAR, które to bity opisują aktywność stronic pamięci PAL w bieżącej i w poprzedzającej kadencji



(jeśli  $r = 0$ , to dana strona była nieaktywna). Jeżeli w pamięci PAL znajduje się stronica wolna o adresie  $i$ , to wykonuje się operacja pobrania żądanego bloku o adresie realnym  $a$ , tj. operacja  $PAL[i] := PAO[a]$ , która poprzedza operację zapamiętania starego bloku o adresie realnym  $p$ , tj. operacja postaci  $PAO[p] := PAL[i]$ . Jeśli natomiast wszystkie strony w PAL są zajęte (wówczas wszystkie bity  $r$  są różne od 0), to wówczas kończy się bieżąca kadencja pracy procesora PE, zeruje się bity  $t$  wszystkich rejestrów RAR oraz dodatkowo bity  $r$  tylko tych rejestrów, których strony były nieaktywne w poprzedniej kadencji.



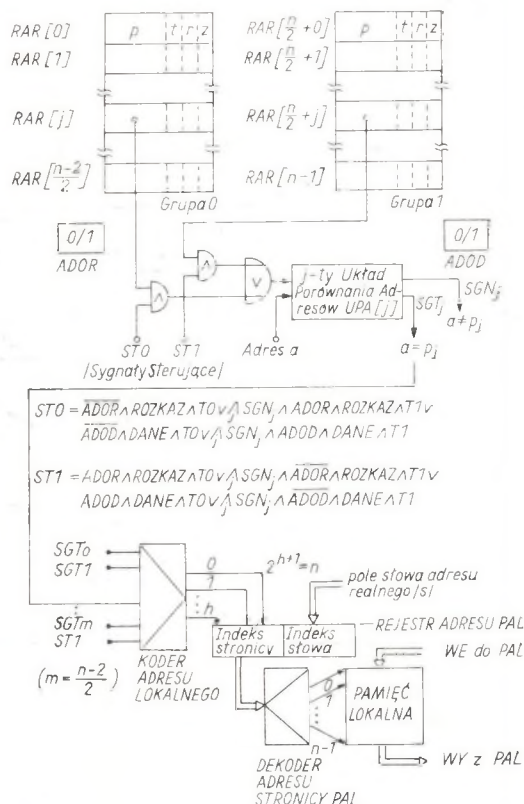
Rys. 2. Algorytm alokacji dynamicznej stronie PAL. Przyjęte oznaczenia:  $r_i = RAR[i]\langle r \rangle$ , tj. stan bitu  $r$   $i$ -tego rejestru RAR;  $z_i = RAR[i]\langle z \rangle$ , tj. stan bitu  $z$   $i$ -tego rejestru RAR;  $t_i = RAR[i]\langle t \rangle$ , tj. stan bitu  $t$   $i$ -tego rejestru RAR;  $p = RAR[i]\langle p \rangle$ , tj. stan pola  $p$   $i$ -tego rejestru RAR;  $PAO[a]$  — stronica pamięci PAO o adresie realnym  $a$ , w której znajduje się żądany blok informacji

### 2.3. Schemat identyfikacji i translacji adresów

Adresy efektywne występujące w programie emulowanym odnoszą się bezpośrednio do komórek pamięci operacyjnej, czyli są to tzw. adresy realne. Z drugiej strony pamięć lokalna procesora emulacji ma własną adresację komórek (słów). Adresy pamięci PAL nazywać będziemy adresami lokalnymi. Zatem blokom informacji, których kopie znajdują się w pamięci PAL przypisuje się zarówno adresy realne, jak i lokalne. W związku z tym w czasie wykonywania programu emulowanego w procesorze PE zachodzi potrzeba szybkiej translacji adresów realnych na lokalne, poprzedzona procedurą identyfikacji adresów realnych. Przez procedurę identyfikacji rozumie się tutaj sprawdzenie

zgodności bieżącego adresu realnego (z dokładnością do strony) z zawartością rejestrów RAR (rys. 3).

Jeśli istnieje taki adres lokalny  $i$ , że bieżący adres realny  $a = \text{RAR}[i] \langle p \rangle$ , to kopia żądanego bloku informacji znajduje się w stronie PAL[ $i$ ]. Jeśli natomiast dla każdego  $i$ , gdzie  $i = 0, 1, \dots, n-1$ ; zachodzi  $a \neq \text{RAR}[i] \langle p \rangle$ , to żądany blok nie znajduje się w pamięci PAL. W tym przypadku uruchamia się procedurę sprowadzania żądanego bloku z PAO, poprzedzoną ewentualnie



Rys. 3. Schemat porównywania i translacji adresów realnych na lokalne

procedurą przechowywania starego bloku (patrz rys. 2). Szczegółowy schemat identyfikacji adresów realnych i tworzenia (kodowania) adresu lokalnego bieżącej strony pokazany jest na rys. 3, gdzie dla oszczędnej realizacji procedury prawie jednoczesnego porównywania adresów zestaw rejestrów RAR został podzielony na dwie grupy rejestrów o adresach  $0, 1, \dots, \frac{n-2}{2}$  oraz

$\frac{n}{2}, \frac{n}{2} + 1, \dots, n-1$ . Dzięki temu schemat porównywania adresów zawiera nie  $n$ , lecz  $\frac{n}{2}$  układów porównania ( $l$ -bitowych). Układ porównywania adresów o nu-

merze  $j$ , tj.  $UPA[j]$ , porównuje w dwóch kolejnych taktach czasowych ( $T_0$  i  $T_1$ ) bieżący adres realny ( $a$ ) z zawartością rejestrów  $RAR[0+j]$  i  $RAR\left[\frac{n}{2}+j\right]$ .

Kolejność porównywania, zadawana sygnałami  $ST_0$  i  $ST_1$ , jest określana wskaźnikami  $ADOR$  ( $AD$ resu  $O$ statniego  $R$ ozkazu) i  $ADOD$  ( $AD$ resu  $O$ statniej  $D$ anej), które to wskaźniki pamiętają numer grupy stronic  $PAL$ , z których pobrano ostatni rozkaz lub z których pobrano ostatnią daną (bądź do których zapisano ostatnią daną). Przyjęta kolejność porównywania zawartości rejestrów  $RAR[0+j]$  i  $RAR\left[\frac{n}{2}+j\right]$ , gdzie  $j = 0, 1, \dots, \frac{n-2}{2}$ , opiera się na założeniu sekwencyjnego charakteru pobierania rozkazów i danych programu wykonywanego.

Zestaw rejestrów  $RAR$  można w konkretnej realizacji podzielić na większą ilość grup niż dwie (stosując zaproponowane rozwiązanie), zależnie od czasu działania stosowanych elementów logicznych w układach porównywania oraz od wymaganego średniego cyklu dostępu do pamięci  $PAL$ , i zmniejszyć jeszcze w ten sposób ilość układów porównywania adresów realnych.

Algorytmy inicjacji programu emulowanego, sposób organizowania transmisji bloków pomiędzy procesorami  $PG$  i  $PE$  oraz sposób zakończenia pracy programu emulowanego — dla rozważanej metody — realizowane są w sposób analogiczny jak dla macierzowej metody emulacji wirtualnej opisanej w pracy [2].

### 3. Zakończenie

Opisana powyżej stronicowa metoda emulacji wirtualnej charakteryzuje się następującymi cechami użytkowymi:

1. Realizacja techniczna omawianej metody emulacji nie wymaga dokonywania zmian zarówno w sprzęcie jak i w systemie operacyjnym komputera emulującego. Zatem można wyposażać w zdolność emulacji nie tylko nowo konstruowane komputery — jak w konwencjonalnych metodach — lecz także już zainstalowane komputery.

2. Opisana metoda pozwala na współbieżne wykonywanie programów emulowanych z programami własnymi komputera emulującego. Przy czym zarówno procesor główny, jak i procesor emulacji mogą być wykorzystane w systemie wieloprogramowym.

3. Założenie decentralizacji mocy obliczeniowej komputera emulującego pozwala na lepsze wykorzystanie sprzętu komputera i na osiągnięcie wyższej łącznej wydajności komputera.

Na zakończenie niniejszej pracy pragniemy poinformować, że porównanie efektywności przedstawionych dwóch metod emulacji wirtualnej (tj. metody macierzowej i stronicowej) jest dokonywane przy założeniu, że emulowanymi programami są programy użytkowe maszyn serii  $ODRA 1300$ , zaś maszyną emulującą jest dowolny komputer Jednolitego Systemu „RIAD I” lub

„RIAD II”. Otrzymane wyniki porównań z przeprowadzonych badań symulacyjnych opisanych metod emulacji komputerowej zostaną opublikowane w innej pracy.

### *Concurrent Computer Emulation*

In the paper a method for transfer of programs written on the level of machine instructions, from a given computer onto other one, is presented. Effectiveness of the proposed algorithm is investigated under assumption that a computer ODRA 1300 plays role of emulated system whereas emulating computer is the system RIAD.

### *Параллельная компьютерная эмуляция*

В статье описываются методы переноса изодного компьютера на другой программ, написанных на уровне машинных инструкций.

Исследуется эффективность предложенных алгоритмов, при условии, что эмулированным компьютером является система ОДРА 1300, а эмулирующим — система РИАД.

### Literatura

- [1] S. Husson, *Microprogramming, Principles and Practices*, Prentice-Hall, Inc., 1970.
- [2] Th. Kamburelis, *Współbieżna emulacja komputerowa*, Podstawy Sterowania, t. 6, z. 4, 1976.
- [3] Th. Kamburelis, Projekty organizacji logicznej emulacji EMC ODRA 1300 w EMC Jednolitego Systemu, MERA-ELWRO (raport wewnętrzny), Wrocław 1976.
- [4] Th. Kamburelis, *Architektura logiczna maszyny cyfrowej ODRA 1305*, MERA-ELWRO, Wrocław 1974.
- [5] Th. Kamburelis, *Architektura logiczna EMC JS*, seria Problemy Informatyki, OBRI, Warszawa 1976.
- [6] Projekt Techniczny EMC JS-1045, MERA-ELWRO (Opracowanie wewnętrzne), Wrocław 1974.
- [7] H. Dryzek, *Organizacja pamięci w systemach liczących. Problemy podziału*, PWN, Warszawa 1970.
- [8] H. Dryzek, *Optymalizacja pamięci wielopoziomowej. Analiza rozwiązań programowych*, PWN, Warszawa 1974.
- [9] L. Belady, *A Study of Replacement Algorithms for Virtual Storage Computer*, IBM Syst. J., vol. 5, no 2, 1966.
- [10] P. Denning, S. Schwartz, *Properties of the Working Set Model*, Comm. ACM, March 1972.
- [11] P. Franaszek, T. Wagner, *Some Distribution Free Aspects of Paging*, J. ACM, vol. 21, January 1974.
- [12] L. Lewis, G. Shedler, *Empirically Derived Micro-models for Sequences of Page Exceptions*, IBM J. Res. Develop., March 1973.
- [13] R. Mattson, *Evaluation Techniques for Storage Hierarchies*, IBM Syst. J., vol. 9, nr 2, 1970.

## INFORMACJE DLA AUTORÓW

Komitet Redakcyjny w celu skrócenia cyklu wydawniczego prosi autorów o opracowywanie materiałów przeznaczonych do druku w Podstawach Sterowania zgodnie z podanymi poniżej wytycznymi:

1. Prace winny być pisane na maszynie jednostronnie, na pojedynczych arkuszach formatu A4, z podwójną interlinią, z marginesem 3,5 cm, z lewej strony. Stronice numerowane. Prace należy nadsyłać w dwóch egzemplarzach.

2. Wzory i oznaczenia należy wpisać czytelnie, używając jedynie liter łacińskich i greckich. Wskaźniki, niżej liter i wykładniki potęg pisać należy dokładnie i wyraźnie.

3. Prace mogą być nadsyłane w jednym z pięciu języków: polskim, angielskim, francuskim, niemieckim i rosyjskim. Każda praca powinna być zaopatrzona w krótkie streszczenie (do 25 wierszy maszynopisu) oraz w obszerniejsze streszczenia (ok. 20% objętości artykułu) w języku angielskim i rosyjskim, jeśli praca jest w języku polskim, natomiast w polskim i rosyjskim w innych przypadkach.

4. Rysunki i wykresy należy wykonać na oddzielnych arkuszach i co najmniej dwukrotnie większe niż mają być w druku. Na każdym rysunku musi być zaznaczony kolejny numer i nazwisko autora. Oprócz tego należy dołączyć wykaz podpisów pod rysunkami.

5. Wszystkie tablice, podobnie jak rysunki, należy wykonać na oddzielnych arkuszach i numerować je kolejno liczbami arabskimi. U góry tablicy podać tytuł objaśniający.

6. Spis literatury należy umieścić na końcu wymieniając w następującej kolejności pierwsze litery imion i nazwisko autora, pełny tytuł dzieła lub artykułu, tytuł czasopisma, tom, numer zeszytu, rok, miejsce wydania oraz numery stron. Pozytcje powinny być ponumerowane. Odsyłacze do literatury w tekście należy oznaczać kolejnymi cyframi umieszczonymi w nawiasach kwadratowych.

7. Na końcu pracy po nazwisku autora powinna być podana nazwa zakładu pracy i adres.

8. Autorowi przysługuje bezpłatnie 25 egzemplarzy odbitek pracy. Dodatkowe egzemplarze autor może zamówić w redakcji na własny koszt przy przesyłaniu korekty swej pracy lub na koszt zakładu pracy.

Uwaga: Autora obowiązuje korekta autorska, którą należy zwracać w ciągu 3 dni pod adresem: Państwowe Wydawnictwo Naukowe, 31-112 Kraków, ul. Smoleńsk 14, Dział Czasopism.

9. Prace nadsyłać należy bezpośrednio do redaktorów na następujące adresy:

z zakresu: Podstawowe problemy struktur  
sterowania

PROF. JERZY BROMIRSKI

Instytut Cybernetyki Technicznej  
Politechnika Wrocławska

ul. Janiszewskiego 11/17, 50-372 Wrocław

z zakresu: Matematyczne podstawy teorii  
sterowania

PROF. CZESŁAW OLECH

Instytut Matematyczny PAN  
Warszawa

ul. Śniadeckich 8

z zakresu: Podstawowe problemy algorytmów  
sterowania i oprogramowanie jednostek centralnych

PROF. STEFAN WĘGRZYN

Zakład Systemów Automatyki Kompleksowej PAN  
ul. Zwycięstwa 2, 44-100 Gliwice

PODSTAWY STEROWANIA

(kwartalnik)

*Warunki prenumeraty czasopisma*

Cena prenumeraty krajowej

rocznie      zł 60.—

półrocznie  zł 30.—

Prenumeratę na kraj przyjmują Oddziały RSW „Prasa—Książka—Ruch” oraz urzędy pocztowe i doręczyciele w terminach:

— do dnia 25 listopada na styczeń, I kwartał, I półrocze roku następnego i cały rok następny,

— do dnia 10 miesiąca poprzedzającego okres prenumeraty.

Jednostki gospodarki uspołecznionej, instytucje i organizacje społeczno-polityczne składają zamówienia w miejscowych Oddziałach RSW „Prasa—Książka—Ruch”.

Zakłady pracy w miejscowościach, w których nie ma Oddziałów RSW oraz prenumeratory indywidualni, zamawiają prenumeratę w urzędach pocztowych lub u doręczycieli.

Prenumeratę ze zleceniem wysyłki za granicę, która jest o 50% droższa od prenumeraty krajowej, przyjmuje RSW „Prasa—Książka—Ruch”, Centrala Kolportażu Prasy i Wydawnictw, ul. Towarowa 28, 00-958 Warszawa, konto PKO 1531-71 w terminach podanych dla prenumeraty krajowej.

Bieżące i archiwalne numery można nabyć lub zamówić we Wzorcowni Wydawnictw Naukowych PAN — Ossolineum — PWN, Pałac Kultury i Nauki (wysoki parter) 00-901 Warszawa oraz w księgarniach naukowych „Domu Książki”.

**Tylko prenumerata zapewnia regularne otrzymywanie czasopisma**

*Information for subscribers*

Published quarterly, one volume a year. A subscription order stating the period of time, subscriber's name and address can be sent to any subscription agent or directly to the Foreign Trade Enterprise ARS POLONA — RUCH, 00-068 Warszawa, Krakowskie Przedmieście 7, P. O. Box 1001, Poland.

Please send payments (annual subscription US \$ 8.0) to the account of ARS POLONA — RUCH, Bank Handlowy S. A., Traugutta 7, 00-067 Warszawa, Poland.