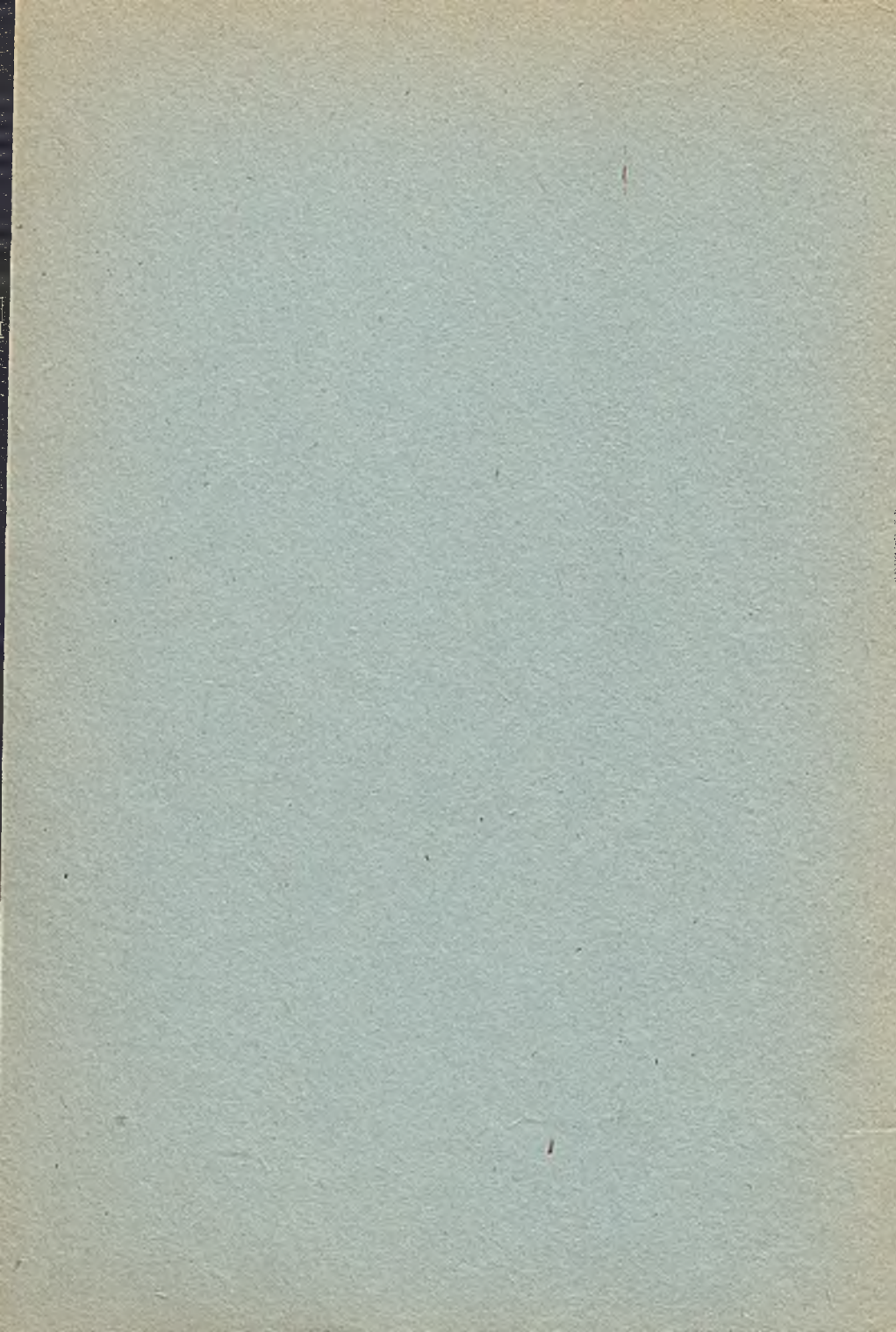


III

**KONFERENCJA UŻYTKOWNIKÓW
MINIKOMPUTERA MERA-400**

GDANSK



INSTYTUT OKRETOWY
POLITECHNIKI GDANSKIEJ

POROZUMIENIE UŻYTKOWNIKOW
MINIKOMPUTERA MERA-400
PRZY POLSKIM TOWARZYSTWIE
CYBERNETYCZNYM
ODDZIAŁ W GDAŃSKU

III KONFERENCJA
UŻYTKOWNIKOW MINIKOMPUTERA
MERA-400

Materiały

Gdańsk, 23-26 października 1985 r.

KOMITET ORGANIZACYJNY KONFERENCJI

Przewodniczący:

Andrzej Braniecki, Instytut Okrętowy PG

Sekretarz naukowy i redaktor materiałów:

Stefan Zieliński, PTC Oddział Gdańsk, Instytut Okrętowy PG.

Sekretarz organizacyjny:

Grzegorz Cerkaski, Instytut Okrętowy PG

Rada Programowa Konferencji - Rada Programowa Porozumienia

Użytkowników Minikomputera MERA-400

1. Andrzej Braniecki /przewodniczący/
2. Jerzy Dżoga
3. Tadeusz Grall
4. Hanna Krzyszczyk
5. Bogdan Machowiak
6. Tomasz Rawiński
7. Andrzej Stoś
8. Marian Waksman
9. Jan Wierzbicki
10. Stefan Zieliński

SPIS TRESCI

WSTEP	7
I. UKŁADY WIELOKOMPUTEROWE	
ZASTOSOWANIE MERY-400 W LOKALNEJ SIECI KOMPUTEROWEJ K. Anzelewicz, Z. Czerniak	10
CELOWOSC PROWADZENIA BADAN NAD BUDOWA SYSTEMOW TELEPRZETWARZANIA B. Machowiak	19
POŁĄCZENIE MINIKOMPUTERA IBM-PC Z MERA-400 R. Tuziński	23
SYSTEM WIELOSTANOWISKOWEJ PRACY POD NOP-EM, JEDNOCZESNEGO PRZESYŁANIA WIELU ZBIOROW PO KILKU ŁĄCZACH DLA MASZYN MERA-400 i ODRA-1305 H. Jusza, A. Tuzińska	26
SICOB'85 J. Pulwarski	34
II. OPROGRAMOWANIE SYSTEMOWE	
METODOLOGIA PROGRAMOWANJA W LOGLANIE A. Salwicki	43
KONCEPCJA ROZWOJU OPROGRAMOWANIA PODSTAWOWEGO MINIKOMPUTERA MERA-400 B. Machowiak	56
ANALIZA CHARAKTERYSTYCZNYCH CECH SYSTEMOW OPERACYJNYCH CROOK-4 I UNIX A. Ziemkiewicz	73

PROBLEMY PRZENOSZENIA OPROGRAMOWANIA UŻYTKOWEGO
Z SYSTEMU SOM-3 NA CROOK-4
J. Wierzbicki, M. Perek 87

DOSWIADCZENIA Z EKSPLOATACJI SYSTEMOW OPERACYJNYCH
CROOK I DSM-1
A. Pisiewicz 91

SYSTEM OPERACYJNY DSM-1
H. Olkowski 96

SYSTEM OPERACYJNY SOM 3.P - WERSJA V.2
M. Guja, K. Jojczyk 114

TRANSLATOR JEZYKA ALGOL-1204 MD DLA MINIKOMPUTERA
MERA-400 Z SYSTEMEM OPERACYJNYM CROOK-4
J. Gocałek, J. Klauziński, R. Zenker 120

III. BAZY DANYCH

PROCESY PROJEKTOWANIA BAZ DANYCH
S. Wrycza 127

PRZEGLAD SYSTEMOW OBSŁUGI I ORGANIZACJI ZBIOROW
DANYCH NA MERA-400
J. Wierzbicki 152

ILUSTRACJA BAZY DANYCH "SINK" NA PRZYKŁADZIE
OBSŁUGI MIĘDZYKRAJOWEGO TURNIEJU KOSZYKOWKI
KOBIEĆ O "PUCHAR BAŁTYKU"
L. Byczkowski 163

ROZWOJ ZASTOSOWAN BAZY DANYCH SELKO
M. Meler-Kapcia 172

ZASTOSOWANIE BAZY DANYCH BD-83 VITRIN POD
SYSTEMEM OPERACYJNYM CROOK-4
M. Perek, J. Wierzbicki 177

IV. SYSTEMY APLIKACYJNE

SYSTEM FINANSOWO-KOSZTOWY NA MINIKOMPUTERZE MERA-400 T. Dyrga	181
SYSTEM PROFIN - PLANOWANIE I ROZLICZANIE FINANSOWE PRZEDSIĘBIORSTWA J. Pulwarski	186
WARIANTOWANIE SEKCJI MIESZKANIOWYCH PRZY ZADANYCH MODUŁACH WIELKIEJ PŁYTY B. Machowiak	199
ZASTOSOWANIE SYSTEMU MERA-400 - CAMAC DO ZBIERANIA I PRZETWARZANIA DANYCH POMIAROWYCH W PRACOWNI CHEMICZNEJ ANALIZY INSTRUMENTALNEJ M. Biernat, R. Radomski	203
WYKORZYSTANIE MINIKOMPUTERA MERA-400 I KANAŁU INTELDIGIT PI DO STEROWANIA W CZASIE RZECZYWISTYM WIELOMA SPEKTROMETRAMI DLA OZNACZANIA SKŁADU CHEMICZNEGO J. Kocój, Z. Werfel	208
MODEL RUCHU POJAZDOW W SIECI SKRZYŻOWAN ZREALIZOWANY W JEZYKU CSL/CROOK-4 R. Konieczny, S. Krawiec	212
ZESTAW PROGRAMOW DO WYGLĄDZANIA I ROZNICZKOWANIA DANYCH POMIAROWYCH M. Radomska, P. Nowak, S. Janicki, R. Radomski	218
OPROGRAMOWANIE DLA METOD OPTIMALIZACJI STATYCZNEJ I STATYSTYKI NA MINIKOMPUTERZE MERA-400 F. Przechowski, S. Zieliński	224

V. KOMUNIKATY I OFERTY

MERITUM I TERMINALEM V24 H. Gruja, K. Jojczyk	240
ZESTAW MAKRODEFINICJI ASSEMBLERA DO WSPÓLPRACY MERA-400-CAMAC W SYSTEMIE OPERACYJNYM SOM-3 L. Czerwosz	242
SYSTEM ANCO J. Gocałek, J. Klauziński, R. Zenker	243
ADAPTACJA PROGRAMU PROBUS W SYSTEMIE OPERACYJNYM CROOK-4 G. Granowicz	244
DOM HANDLOWY NAUKI - "PROGRAMEX"	245
AMEPOL	246
MERCOMP	251
OFERTA FIRMY "ZEKOM"	264
SPOŁKA USŁUG INFORMATYCZNYCH "INFORMATYKA"	271
ZAKŁAD ELEKTRONICZNY MIESZKO CIEPLINSKI	272
ZAKŁAD SYSTEMOW CYFROWYCH "DIGITEX"	273
TOBI	274
LISTA UCZESTNIKÓW III KONFERENCJI UŻYTKOWNIKÓW MINIKOMPUTERA MERA-400	275
LISTA CZŁONKÓW POROZUMIENIA UŻYTKOWNIKÓW MINIKOMPUTERA MERA-400	285

WSTĘP

III Konferencja Użytkowników Minikomputera MERA-400 odbyła się w dniach 23-25 października 1985 r. w Gdańsku w budynku NOT-u oraz w dniach 25-26 w Instytucie Okrętowym Politechniki Gdańskiej.

Konferencja zgromadziła około 180 uczestników i obserwatorów.

Na Konferencji wygłoszono 42 referaty i komunikaty w następujących sesjach:

1. Teoretyczne podstawy systemów informatycznych.
2. Przyszłość sprzętowa użytkowników MERY-400.
3. Układy wielokomputerowe.
4. Przenośność oprogramowania.
5. Bazy danych.
6. Systemy aplikacyjne.

W drugim dniu Konferencji odbyło się zebranie plenarne członków Porozumienia Użytkowników Minikomputera MERA-400, na którym przedyskutowano i przyjęto sprawozdanie z dotychczasowej działalności Porozumienia oraz planowane zadania na rok 1986.

W trakcie zebrania dokonano wyborów uzupełniających do Rady Programowej Porozumienia.

Aktualny skład Rady jest następujący:

1. Andrzej Braniewski /przewodniczący/
2. Jerzy Dżoga
3. Tadeusz Grall
4. Bogdan Machowiak
5. Tomasz Rawiński
6. Jadwiga Stecura
7. Wojciech Szanser
8. Marian Wakeman
9. Jan Wierzbicki
10. Stefan Zieliński

W podsumowaniu Konferencji i dyskusji na zebraniu plenarnym można stwierdzić:

- w dziedzinie oprogramowania użytkowego podstawowym problemem pozostaje przenośność systemów pomiędzy maszynami MERA-400 o różnych systemach operacyjnych oraz na inne komputery, które mogą pojawić się u użytkowników MERY-400.

- w dziedzinie sprzętu występuje zapotrzebowanie na rozbudowę istniejących systemów MERA-400 o procesory komunikacyjne oraz duże pamięci dyskowe i operacyjne. Występuje również zapotrzebowanie na budowę sieci komputerowych z wykorzystaniem MERY-400.
- w ramach Porozumienia Użytkowników powstanie sekcja zastosowań przemysłowych. Zorganizowanie takiej sekcji podjął się W.Badura z HSO "KARA" w Piotrkowie Trybunalskim.
- IV Konferencja Użytkowników Minikomputera MERA-400 odbędzie się w październiku 1986 r. w oparciu o bazę noclegową ośrodków wczasowych zlokalizowanych w Gdańsku. Przewiduje się wydanie materiałów przed Konferencją.
Materiały zawierają referaty, komunikaty i oferty prezentowane na Konferencji. Włączono do nich także, ze względu na interesującą tematykę, dwa referaty dr J.Pulwarskiego, a także wyniki ekspertyz mgr inż. B.Machowiaka /"Koncepcja rozwoju oprogramowania podstawowego minikomputera MERA-400"/ i mgra inż. J.Wierzbickiego /"Przegląd systemów obsługi i organizacji zbiorów danych na MERA-400"/.

I UKŁADY WIELOKOMPUTEROWE

mgr inż. Krzysztof ANZELEWICZ
mgr inż. Zbigniew CZERNIAK
Politechnika Gdańska
Instytut Okrętowy

ZASTOSOWANIE MERY-400 W LOKALNEJ SIECI KOMPUTEROWEJ

Omówiono historię powstania LAN. Przedstawiono standardy IEEE 802 LAN. Pokazano zależności występujące między Modelem 802 LAN a Modelem Połączeń Systemów Otwartych wg ISO. Podano podstawowy hardware niezbędny do wykonania połączenia Komputer/Sieć. Przedstawiono dotychczasowe prace związane z zastosowaniem MERY-400 w LAN.

1. WSTĘP

Szybki rozwój konstrukcji urządzeń komputerowych i technologii wytwarzania układów scalonych VLSI doprowadził do upowszechnienia taniego sprzętu komputerowego w instytutach naukowo-badawczych, biurach i zakładach przemysłowych. Przeszkodą w prawidłowym wykorzystaniu nagromadzonego sprzętu jest jego niekompatybilność. Uniemożliwia to wymianę oprogramowania, dostęp do bazy danych czy wykorzystanie drogich urządzeń zewnętrznych typu dyski, drukarki, plotery należące do jednego systemu komputerowego, przez drugi system takich urządzeń nie posiadający. Od połowy lat siedemdziesiątych prowadzi się prace nad połączeniem sprzętu komputerowego w Lokalną Sieć Komputerową /ang. Local Area Network - LAN/ umożliwiającą bezpośrednio i szybką wymianę informacji między wieloma urządzeniami zlokalizowanymi w bliskiej odległości. Pierwszą sieć pod nazwą "Ethernet" uruchomiła w 1975 roku firma Xerox. W 1976 roku firma Datapoint wypuściła na rynek sieć pod nazwą "ARCnet". Na Uniwersytecie Cambridge w 1978 roku zbudowano sieć pod nazwą "Cambridge Ring". Następne lata przyniosły bardzo wiele nowych sieci lokalnych o różnych zasadach pracy. Aktualnie jest około 200 producentów LAN różnego typu. W związku z tak dużą ilością rodzajów LAN zaistniała potrzeba standaryzacji. Od 1980 roku działa powołany przez IEEE /ang. Institute of Electrical and Electronics Engineers/ Komitet 802 do spraw standaryzacji LAN. Jednocześnie od 1979 roku ISO /ang. International Standard Organization/ propaguje Warstwowy Model Połączeń Systemów Otwartych /ang. Layer Model for Open Systems Interconnection-OSI/ISO/.

Komitet 802 IEEE LAN w 1983 roku zaproponował trzy standardy dla LAN, które jednocześnie uwzględniają zalecenia OSI/ISO. Rok 1984 przyniósł nowy mocny akcent na drodze rozwoju standaryzacji LAN w postaci MAP /ang. Manufacturing Automation Protocol/ firmy General Motors. W listopadzie br. przewiduje się przedstawienie LAN bazującego na protokole MAP łączącego urządzenia komputerowe sieciem szeregowym czołowych firm komputerowych takich jak: IBM, Hewlett-Packard, DEC, Honeywell, Intel, Motorola, NCR, Siemens.

2. DEFINICJA LAN

Za LAN uważa się bezpośrednią komunikację wielu urządzeń komputerowych działających na ograniczonym obszarze, połączonych wspólnym medium fizycznym umożliwiającym szybką transmisję szeregową. Obszar działania LAN wynosi od 1 do 10 km tzn. jeden lub kilka pobliskich budynków. Prędkość transmisji od 1 do 20 Mbs. Ilość urządzeń komputerowych mających bezpośredni dostęp do sieci rzędu 200 lub więcej. Prawdopodobieństwo błędu transmisji rzędu 10^{-6} przy zastosowaniu 32 bitowego kodu kontrolnego. Medium fizyczne to ekranowana skrętka, kabel koncentryczny lub światłowód. Typowym dla LAN jest istnienie w sieci wyspecjalizowanych punktów obsługi zbiorów /ang. File Server/ realizacji wydruków /ang. Print Server/ itp. obsługujących wszystkich użytkowników sieci. Dzięki temu każdy użytkownik LAN nie musi posiadać własnej drukarki czy pamięci dyskowej a może korzystać ze wspólnych dla całej sieci. Oczywiście w sieci może istnieć kilka punktów obsługi tego samego typu np. punktów obsługi zbiorów zawierających pamięci dyskowe i taśmowe.

3. STANDARDY IEEE 802 LAN

Komitet IEEE 802 powołał trzy podkomitety, które opracowały następujące standardy dla LAN:

- IEEE 802.3, sieć szynowa ze śledzeniem nośnej i detekcją interferencji /ang. Bus with Carrier Sense Multiple Access/Collision Detection - CSMA/CD /,
 - IEEE 802.4, sieć szynowa z przesyłaniem pozwolenia / ang. Token Bus/,
 - IEEE 802.5, sieć pętlowa z przesyłaniem pozwolenia /ang. Token Ring/.
- Standardy IEEE 802 zalecają dwie topologie dla LAN: szynową i pętlową oraz dwie metody dostępu CSMA/CD i Token passing. W topologii szynowej

medium fizyczne jest otwarte na obu końcach stanowiąc swego rodzaju szynę komputerową, do której dołączone są wszystkie urządzenia komputerowe sieci. Topologia pętlowa charakteryzuje się tym, że sieć składa się z odcinków medium fizycznego łączących poszczególne urządzenia między sobą w całość tworząc zamkniętą pętlę. IEEE definiuje dwie metody dostępu do sieci:

- CSMA/CD,
- przesyłanie pozwolenia /ang. Token passing/.

3.1. METODA CSMA/CD

Wszyscy użytkownicy sieci mają niezależny dostęp do sieci i współzawodniczą między sobą by go uzyskać. Przed zapoczątkowaniem transmisji użytkownik nasłuchuje sieć, by sprawdzić czy nikt inny nie nadaje. Jeśli sieć jest wolna nadaje, a gdy zajęta to odczeka pewien okres czasu i ponawia próbę transmisji. W czasie nadawania również nasłuchuje, by w przypadku wykrycia kolizji natychmiast zakłócić sieć by odbiorca transmisji odebrał to jako przerwanie transmisji. Metoda dostępu CSMA/CD jest używana w powiązaniu z topologią szynową co przedstawia rys. 1.



Rys. 1. Protokół dostępu CSMA/CD powoduje, że jedna stacja nadaje a reszta odbiera jednocześnie oczekując na uzyskanie dostępu do sieci

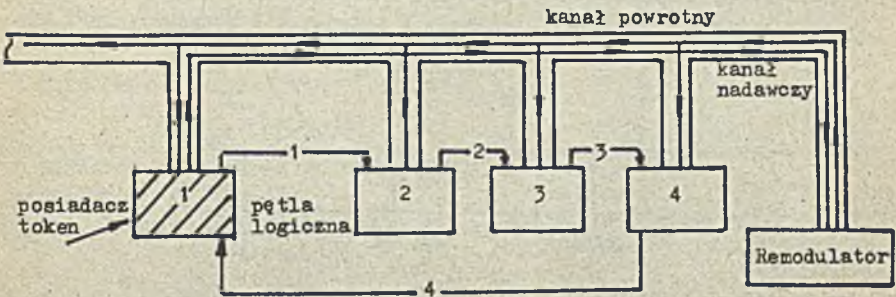
Zaletą CSMA/CD jest szybkość otrzymania dostępu przy niewielkim obciążeniu sieci. Czas propagacji sygnału przez medium fizyczne jest jej wadą i powoduje ograniczenie długości sieci do 2500 m.

3.2. METODA TOKEN PASSING

Specjalna ramka zwana pozwoleniem jest przesyłana od użytkownika do użytkownika w ustalonej kolejności. Tylko użytkownik otrzymujący pozwolenie może nadawać. Użytkownik przekazuje pozwolenie następcy gdy nie ma nic więcej do transmisji lub gdy upłynął czas przez jaki może trzymać pozwolenie. Token passing może być stosowane zarówno dla sieci pętlowej jak i szynowej.

3.2.1. TOKEN BUS

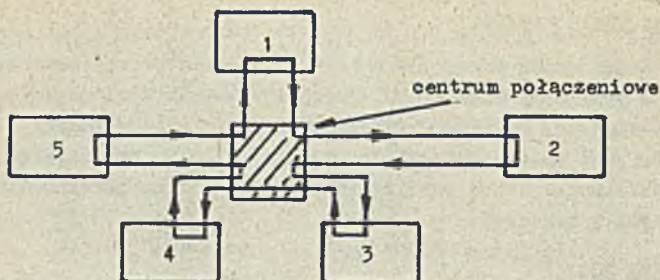
Ramka jest przesyłana w tzw. pętli logicznej wynikającej z adresów poszczególnych użytkowników. Każdy z nich musi znać adres swojego poprzednika jak i następcy w pętli logicznej. Przedstawia to schematycznie rys. 2.



Rys. 2. Użytkownicy sieci wielokanałowej otrzymują pozwolenie zgodnie z łączącą ich pętlą logiczną. Sygnał od nadawcy idzie do remodulatora, który musi go zmienić na inną częstotliwość.

3.2.2. TOKEN RING

Ramka jest przesyłana od użytkownika do użytkownika. Każdy z nich retransmituje ją dalej aż do osiągnięcia adresata, który ją kopiuje i przesyła dalej tak by dotarła do nadawcy. Pełna rotacja ramki poprzez pętlę łączącą użytkowników jest częścią składową tej metody dostępu. W sieciach pętlowych stosuje się centra połączeniowe w celu zwiększenia niezawodności sieci. Typową sieć pętlową przedstawia rys. 3.



Rys. 3. Pozwolenie jest przesyłane od użytkownika do użytkownika

W sieciach z dostępem Token passing łatwiej jest określić czas dostępu poszczególnych użytkowników do sieci jak i stosować priorytety niż w sieciach z dostępem CSMA/CD. Tabela 1 przedstawia standardy IEEE 802.

Tabela 1

Standard	Metoda dostępu	Topologia	Medium fizyczne	Szybkość transmisji
IEEE 802.3	CSMA/CD	Szynowa	Kabel konc. jednokanałowy	10 Mbs
IEEE 802.4	Token passing	Szynowa	Kabel konc. jednokanałowy Kabel konc. wielokanałowy	1, 5, 10 Mbs 1, 5, 10, 20Mbs
IEEE 802.5	Token passing	Pętlowa	Skrętka ekr. Światłowód	1, 4 Mbs 20 Mbs

4. STANDARDY IEEE 802 LAN A WARSTWOWY MODEL OSI/ISO

Model IEEE 802 LAN podzielił warstwę komunikacji OSI/ISO na dwie podwarstwy:

- połączeń logicznych /ang. Logical Link Control - LLC/,
- kontroli dostępu /ang. Medium Access Control - MAC/.

Podwarstwa LLC, standard IEEE 802.2, jest wspólna dla wszystkich typów LAN i zapewnia przesyłanie ramek zgodnie z protokołem komunikacyjnym HDLC /ang. High Level Data Link Control/. Ramka tworzona jest przez

dodanie do informacji właściwej flag na obu jej końcach, pola adresowego, pola sterującego i pola kontroli błędów. Ramkę przedstawia rys. 4. Podwarstwa MAC w powiązaniu z warstwą transmisji jest tym co różni poszczególne typy LAN. Zależności występujące pomiędzy IEEE 802 a OSI/ISO przedstawia rys. 5.

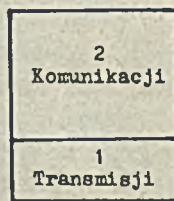
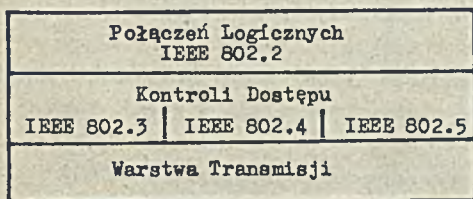


A-pole adresowe, B-pole sterujące, C-informacja właściwa, D-pole k.błędów

Rys. 4. Ramka HDLC

Podwarstwa IEEE 802 LAN

Warstwa ISO



Rys. 5. Zależności pomiędzy standardami IEEE 802 a modelem OSI/ISO

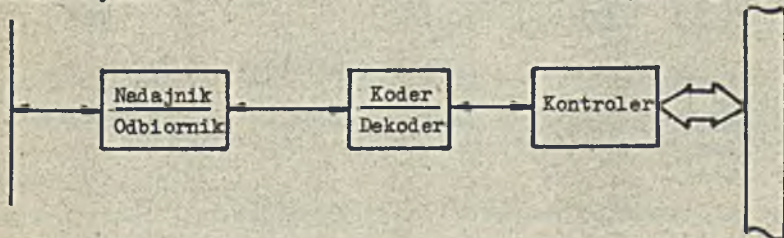
5. HARDWARE LAN

Trzy podstawowe interfejsowe układy scalone stanowią hardware LAN: kontroler, koder/dekoder, nadajnik/odbiornik.

Schematyczny sposób połączeń w/w elementów przedstawia rys. 6.

Medium fizyczne

Szyna komputerowa

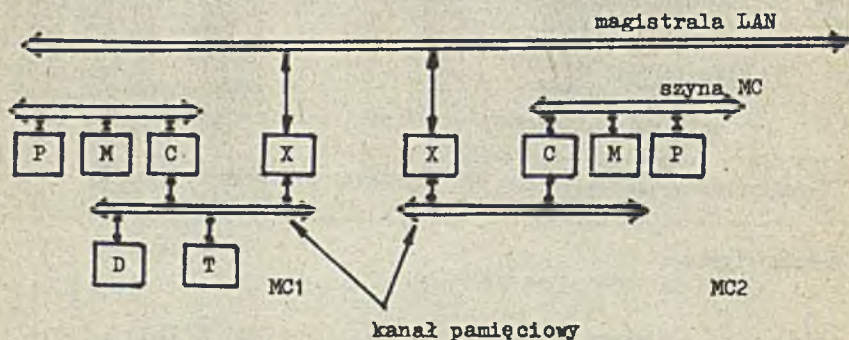


Rys. 6. Podłączenie komputera do sieci

6. MERA-400 W LAN TYPY MAGISTRALOWEGO /SZYNOWEGO/

Eksploatacja w jednym ośrodku obliczeniowym kilku maszyn MERA-400 pracujących w systemie wielodostępnym stwarza wiele problemów organizacyjnych zwłaszcza przy wykorzystaniu takich zasobów jak pamięci dyskowe, pamięci taśmowe, drukarki itp. Rozwiązaniem byłby system, w którym każdy użytkownik miałby dostęp do wszystkich zasobów niezależnie od tego, z którą maszyną jest połączony. Korzystanie z pamięci dyskowych innych maszyn nakłada wymagania na szybkość tego połączenia. Połączenie poprzez interface V24 jest absolutnie niezadowolające ze względu na zbyt małą prędkość transmisji. Narodziła się więc koncepcja zbudowania szybkiego łącza międzykomputerowego o architekturze bazującej na magistralnym LAN z dostępem typu CSMA/CD. Z uwagi na ograniczone możliwości techniczne jak i czas realizacji przedsięwzięcia przyjęto niemierzegowa a równoległy sposób transmisji w magistrali całych słów 16-bitowych. Fizycznie magistralą jest typowy kabel interfejsu pamięciowego MERY-400 łączący poszczególne maszyny pracujące w sieci. Ogranicza to zasięg sieci do kilkunastu metrów.

Urządzenie sprzęgające wykonano jako jednostką sterującą pracującą w kanale pamięciowym MERY-400. Połączenie maszyn z magistralą LAN przedstawia rysunek 7.



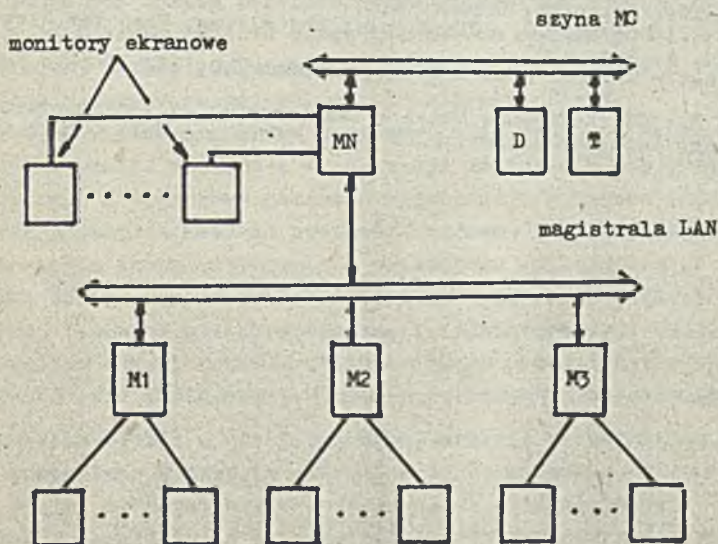
P- procesor, M- pamięć, D- pamięć dyskowa, T- pamięć taśmowa, C-kanal pamięciowy, X- urządzenie sprzęgające z magistralą LAN

Rys. 7. Połączenie maszyn MERA-400 do magistrali LAN

Z punktu widzenia pojedynczej maszyny adresowanie innej MERY-400 nie różni się od adresowania każdego urządzenia pracującego w kanale pamięciowym. W magistrali sprzęgającej przewidziano 4-bitowy adres tak, że w ten sposób możliwa jest współpraca 16 MER-400.

7. SOFTWARE POŁĄCZENIA LAN

Najprostszym wykorzystaniem tak zaprojektowanej sieci może być system hierarchiczny składający się z jednej maszyny nadrzędnej i wielu podległych. Przedstawia to rys. 8. Do maszyny nadrzędnej podłączono wszystkie wspólne urządzenia jak pamięci taśmowe, pamięci dyskowe, drukarki itp. Maszyny podległe mogą pracować w ogóle bez pamięci dyskowych lub posiadać tylko pamięci zbiorów roboczych.



Rys. 8. Zastosowanie LAN z maszyną nadrzędną MN

Maszyny podległe wyposażone są w standardowy system operacyjny CROOK-4. Odwołanie do pamięci dyskowej kierowane jest poprzez magistralę LAN do maszyny nadrzędnej wyposażonej w specjalny podsystem obsługi sieci. Podsystem ten przyjmuje zgłoszenie z sieci, organizuje żadaną transmisję między dyskiem a własną pamięcią a następnie między pamięcią własną a pamięcią maszyny podległej. Podobnie realizowany jest dostęp do innych urządzeń. W celu zwiększenia zasięgu sieci przewidziane jest zastosowanie transmisji szeregowej po kablu koncentrycznym z zastosowaniem Token passing.

LITERATURA

1. Anzelewicz K.: Kierunki rozwoju lokalnych sieci komputerowych, Konfer. Sieci komputerowe, 22-23, Paźdz. 1985, Wrocław
2. Bobcow A., Czerniak Z., Nikodemski M.: System operacyjny CROOK-4 dla minikomputera MERA-400, Wydawnictwo Politechniki Gdańskiej 1985
3. Clark D.D., Pogram K.T., Reed D.P.: An introduction to Local Area Networks, Proc. IEEE, Vol. 66, No. 11, Nov. 1978
4. Data Processing - Open Systems Interconnection - Basic Reference Model, Computer Networks, Vol. 5, No. 2, April 1981
5. IEEE Project 802 - Local Area Network Standards, Draft E IEEE Standard 802.5, IEEE, New York, Aug. 1984
6. Metcalfe R.M., Poggs D.R.: Ethernet: Distributed Packet Switching for Local Computer Networks, Commun. ACM, Vol. 19, No. 7, July 1976
7. Saltzer J.H., Pogram K.T., Clark D.D.: Why a ring, Computer Networks Vol. 7, No. 4, Aug. 1983
8. The Ethernet - A Local Area Network, Version 2.0, DEC, Intel, Xerox, Nov. 1982
9. Zimmerman H.: OSI Reference Model - the Isomodel of Architecture for Open Systems Interconnection, IEEE Trans. Commun., Vol. COM-28, No. 4, April 1980

Biurow Projektów Budownictwa Wiejskiego
Poznań, ul. Piekary 17

Inż. Bogdan Machowiak
tel. 33-05-81 wew. 569

CELOWOŚĆ PROWADZENIA BADAŃ NAD BUDOWĄ SYSTEMÓW TELEPRZETWARZANIA

Analizując najważniejsze tendencje rozwojowe informatyki na świecie w ostatnim dziesięcioleciu, te które będą wyznaczać dominujący wpływ na jej rozwój w latach osiemdziesiątych i dziewięćdziesiątych należałoby zwrócić uwagę głównie na rozwój techniki mikroprocesorowej i rozwój systemów informatycznych oraz co z tym jest ściśle związane rozwój usług teleinformatycznych.

Technika mikroprocesorowa urzeczywistnia postulat daleko posuniętej miniaturyzacji sprzętu i stopniowego zacierania się granic między fizycznym a logicznym poziomem organizacji systemów informatycznych, a także obniżce kosztów systemów, wzrostu ich niezawodności, udostępniania innym użytkownikom potencjału obliczeniowego dostępnego tylko lokalnie w ośrodkach obliczeniowych posiadających maszyny cyfrowe. Technika mikroprocesorowa i teleinformatyka umożliwi wirtualny dostęp do dużych zasobów obliczeniowych przez lokalne ośrodki użytkowników nie posiadających maszyn cyfrowych i oprogramowania.

Dodatkowo, jeśli w sieci teleprzetwarzania uczestniczyć będą Ośrodki wyposażone w maszyny cyfrowe i oprogramowanie, wpłynąć to będzie na rozwój oprogramowania systemowego i aplikacyjnego.

Problemy przejścia od tradycyjnego trybu przetwarzania do trybu zdalnego i rozproszonego w oparciu o środki teleinformatyki jest problemem wyboru między argumentami ZA i PRZECIW.

Do głównych argumentów ZA można zaliczyć:

1. Możliwość realizacji zadań informatycznych, których realizacja w systemie tradycyjnym jest nieopłacalna i nieosiągalna.
2. Szerszy zakres terytorialny systemu informatycznego i możliwość pozyskania nowych użytkowników dzięki mniejszym nakładom inwestycyjnym niż instalacja nowego systemu komputerowego wraz z oprogramowaniem.

3. Możliwa do osiągnięcia lepsza jakość usług informatycznych i większa ich atrakcyjność zarówno dla użytkowników jak i dystrybutorów systemów.
4. Możliwość pełniejszego wykorzystania zasobów fizycznych i logicznych oraz możliwość korzystania z zasobów obcych
5. Ogólnie większa konkurencyjność usług informatycznych.
Natomiast istotne argumenty PRZECIW to:
 1. Dodatkowe nakłady inwestycyjne,
 2. Zwiększone problemy zapewnienia wewnętrznej spójności technicznej, logicznej i informacyjnej systemu teleprzetwarzania.
 3. Problemy związane z gwarancją ochrony danych użytkowników.
 4. Bardziej złpny sposób rozliczenia usług teleprzetwarzania.

Ponadto, wobec dużej ilości na rynku różnych mini i mikrokomputerów z procesorami 8 i 16 bitowymi istnieje duża potrzeba korzystania z istniejącego oprogramowania maszyn cyfrowych średniej i dużej mocy obliczeniowej oraz bazy danych. W tym aspekcie nieopłacalne a wręcz bardzo często technicznie niemożliwe jest przeniesienie takiego istniejącego oprogramowania z maszyn średniej i dużej mocy obliczeniowej na różnego rodzaju komputery osobiste.

Praktycznie uzasadnioną alternatywą jest praca tych komputerów osobistych jako inteligentne końcówki w sieci teleprzetwarzania i możliwość korzystania z zasobów programowych, bez danych Ośrodków posiadających komputery średniej i dużej mocy obliczeniowej.

Taki opisany sposób pracy komputera osobistego w sieci teleprzetwarzania nie będzie wykluczał głównych zalet komputerów osobistych.

Najważniejszą sprawą w budowie systemów teleinformatyki są łącza telekomunikacyjne, z uwagi na brak łączy "dzierżawionych" z resortu łączności i wysoki koszt budowy własnych łączy/ jest to możliwe tylko w kilku resortach/ jedyną alternatywą jest budowa sieci teleinformatyki na bazie komputowanej sieci telefonicznej. Dodatkowym pozytywnym aspektem tej alternatywy jest powszechność ogólnokrajowej sieci telefonicznej.

Potrzebę budowy sieci teleinformatyki zgłaszają zarówno użytkownicy - właściciele systemów MERA-400 jak i użytkownicy indywidualni systemów MERA - 400

Jak do tej pory brak jest WZORCA umożliwiającego budowę takiej sieci teleinformatyki w ramach Ośrodków posiadających MERY-400 na łączach sieci telefonicznej komputowanej.

Taki stan rzeczy spowodowany jest następującymi czynnikami.

1. Budowa wzorcowej sieci teleinformatyki jako praca badawcza przekracza możliwości organizacyjne indywidualnego ośrodka dysponującego małymi zasobami ludzkimi.
2. Trudności ekonomiczno-finansowe indywidualnego ośrodka w wypadku finansowania prac badawczych - gdzie efekt końcowy jest nieznan.
3. Konieczność współdziałania w takich nowatorskich pracach kilku silnych merytorycznie Ośrodków posiadających emc MERA-400 oraz konieczny współdziałanie przedstawiciela konstruktora i producenta urządzeń teletransmisji danych.

Budowa WZORCA sieci teleinformatyki nie może ograniczać się tylko do zrealizowania urządzenia zdolnego do przesyłania danych lecz uzyskanie możliwości efektywnego korzystania z takiego urządzenia. Wobec tego, w budowie WZORCA muszą uczestniczyć osoby, które opracują odpowiednie oprogramowanie w celu efektywnego przesyłania danych w sieci teleinformatyki na łączach telefonicznych komputowanych.

Przedsięwzięcie budowy WZORCA sieci teleinformatyki jest możliwe i celowe w ramach Porozumienia Użytkowników Minikomputera MERA-400.

Celem takiego przedsięwzięcia byłoby zbudowanie WZORCA teleinformatyki, jego wszechstronne przetestowanie oraz sporządzenie opracowania możliwości budowy sieci teleinformatyki w następujących aspektach:

1. Technicznych - jakie urządzenia należałoby posiadać
2. Ekonomicznych - jaka jest niezbędna dyspozycja środków inwestycyjnych.
3. Organizacyjnych - wyniki testów WZORCA w różnych konfiguracjach logicznych.

Zakresem tych prac o charakterze typowo badawczym budowy WZORCA sieci teleinformatyki na bazie emc MERA-400 byłyby następujące etapy:

1. Budowa wzorcowego łącza teletransmisji danych na łączu telefonicznym komutowanym sieci okręgowej i międzymiastowej przy użyciu modemów produkcji Zakładów Teleinformatycznych -TELKOM TELETRA w Poznaniu. Nieodzowny jest tutaj udział konstruktorów modemów celem ich dostosowania do emc MERA-400.
2. Wykonawstwo odpowiedniego oprogramowania systemowego i użytkowego po okresie pierwszego testowania.
Niezbędna obecność w tym etapie prac osób zajmujących się zagadnieniami teleinformatyki oraz oprogramowania maszyn cyfrowych pracujących w sieciach teleinformatyki.
3. Przeprowadzenie wszechstronnych testów jakości transmisji danych dla różnych urządzeń końcowych, różnej prędkości przesyłania danych oraz różnych układów logicznych pracy sieci teleinformatyki / wsadowa, interakcyjna/
4. Sporządzenie opracowania obejmującego wyżej omawiane aspekty oraz udostępnienie tego opracowania członkom Porozumienia Użytkowników Minikomputera MERA - 400.

mgr inż. Ryszard Tuziemski
Gdańsk

POŁĄCZENIE MIKROKOMPUTERA IBM-PC Z MERA-400

W ostatnim czasie pojawiła się możliwość zakupu na rynku polskim szeregu modeli mikrokomputerów kompatybilnych z komputerem osobistym IBM-PC. Mikrokomputery takie oferują m.in. firmy Computex, Impol-II, Emix, Karen i wiele innych.

Podstawowe dane techniczne tych komputerów są następujące:

- procesor Intel 8088 /16 bitowa organizacja wewnętrzna, 8-bitowa zewnętrzna szyna danych/,
- możliwość dołączenia koprocesora numerycznego Intel 8087,
- pamięć operacyjna o pojemności do 640 KB,
- 2 jednostki pamięci na dyskach elastycznych 5 1/4 o pojemności po 360 KB,
- możliwość dołączenia dysku Winchester o pojemności 10, 20 lub 30 MB,
- monitor ekranowy monochromatyczny o rozdzielczości w trybie znakowym 25 wierszy po 80 znaków lub w trybie graficznym 200 linii po 640 punktów,
- interfejs szeregowy RS 232 c /V24/,
- interfejs równoległy CENTRONICS

W sumie jest to mikrokomputer o dużych możliwościach obliczeniowych, zwłaszcza w konfiguracji z koprocesorem numerycznym i dyskiem typu Winchester. Z uwagi na posiadanie przez niego interfejsu V24 możliwe było dołączenie go do minikomputera Mera-400 w charakterze inteligentnego terminala.

Połączenie mikrokomputera CS 88 PC /odpowiednik IBM PC/ z minikomputerem Mera-400 zrealizowano przyjmując założenie o jego pełnej wymiarowości z terminalem /monitorem ekranowym lub DZM - 180 KSR/ komputera Mera-400. W praktyce oznacza to, że komputer IBM PC można dołączyć do systemu Mera-400 za pomocą standardowego kabla monitora ekranowego i standardowej jednostki sterującej UZ DAT w miejsce jednego z monitorów. Dołączony w ten sposób IBM PC staje się inteligentnym terminalem minikomputera Mera-400. Terminal ten realizuje dwa tryby pracy:

- tryb monitora ekranowego,
- tryb przesyłania zbiorów.

O aktualnie realizowanym trybie pracy użytkownik jest informowany za pomocą wyświetlanej w dolnej części ekranu linii statusu.

Mikrokomputer IBM PC dołączony do minikomputera Mera-400 rozpoczyna swoje działanie w trybie monitora ekranowego. W trybie tym operator może korzystać ze wszystkich zleceń systemu operacyjnego minikomputera Mera-400, wprowadzać lub poprawiać zbiory źródłowe, uruchamiać programy - a więc pracować tak jak na typowym terminalu.

W trybie monitora ekranowego operator ma również możliwość zmiany trybu pracy na tryb przesyłania zbiorów. Przełączenie rodzaju pracy następuje po wciśnięciu jednego z klawiszy funkcyjnych mikrokomputera IBM PC.

Przed rozpoczęciem transmisji zbioru pomiędzy komputerami Mera-400 i IBM PC użytkownik musi określić kierunek przesyłania oraz podać nazwy zbiorów, pomiędzy którymi ma nastąpić przesyłanie. Sprawdzana jest wówczas dostępność obydwu zbiorów i w przypadku wystąpienia błędu podczas stwierania zbiorów użytkownik

Jest o tym informowany odpowiednim komunikatem, zaś mikrokomputer IBM PC zmienia swój tryb pracy na tryb monitora ekranowego. W czasie trwania transmisji zbioru na ekranie wyświetlany jest licznik przesyłanych rekordów, co pozwala śledzić przebieg przesyłania. Koniec transmisji sygnalizowany jest komunikatem podającym liczbę przesyłanych rekordów oraz czas transmisji, po czym następuje automatyczny powrót do trybu monitora ekranowego. Podczas trwania transmisji zbioru użytkownik ma możliwość przerwania jej w dowolnym momencie i powrotu do trybu monitora ekranowego.

W dotychczasowej eksploatacji połączenia mikrokomputera IBM PC z minikomputerem Mera-400 oprócz wykorzystania go jako dodatkowego terminala systemu Mera-400 głównym zastosowaniem okazało się przesyłanie źródłowych wersji programów z Mery-400 w celu ich adaptacji na IBM PC. Przesłano w ten sposób ponad 20. tys. rekordów programów w Basicu i Fortranie, co umożliwiło ich bardzo szybkie uruchomienie na mikrokomputerze IBM PC. Przesyłało również zbiory znakowe z IBM PC do Mery-400 w celu ich wydrukowania na drukarce, ponieważ brak jej było w zestawie mikrokomputera.

Następnym zastosowaniem o niewykorzystanych dotychczas możliwościach jest przekształcenie mikrokomputera IBM PC w inteligentny terminal graficzny systemu Mera-400. Z uwagi na rozdzielczość ekranu 640x200 punktów, możliwość dołączenia monitora kolorowego, drukarki o możliwościach graficznych oraz pisaków X-Y można go wykorzystywać do przetwarzania i graficznego prezentowania wyników obliczeń wykonywanych na Merze-400.

mgr inż. Henryk Jusza
mgr inż. Anna Tuziemska
Gdańsk

**SYSTEM WIELOSTANOWISKOWEJ PRACY
POD MOP-em, JEDNOCZESNEGO PRZESYLANIA WIELU ZBIORÓW
PO KILKU ŁĄCZACH DLA MASZYN MERA-400 I ODRA-1305**

Prezentowany niżej system teletransmisji umożliwia użytkownikom minikomputera Mera-400 działającego pod systemem operacyjnym CROOK-4 jednoczesne korzystanie ze wszystkich możliwości oferowanych przez komputer Odra-1305 działający pod systemem operacyjnym GEORGE-3.

1. PODSTAWOWE ZAŁOŻENIA SYSTEMU

- 1/ Maszyny Mera-400 i Odra-1305 połączone są za pośrednictwem jednego lub większej liczby fizycznych kanałów łączności. Na fizyczny kanał łączności składają się dwa modemy połączone linią telefoniczną komutowaną lub dzierżawioną. Fizyczny kanał łączności dołączony jest z jednej strony do odpowiedniej jednostki sterującej kanału znakowego minikomputera Mera-400, zaś z drugiej - do odpowiedniej linii skanera systemu komputerowego Odra-1305.
- W chwili obecnej ze względu na możliwości sprzętowe minikomputera Mera-400 fizyczny kanał łączności realizuje transmisję asynchroniczną z prędkościami wynikającymi z ograniczeń narzuconych przez modemy i charakterystyki przenoszenia linii telefonicznych - praktycznie do 1200 bpdów. Po uruchomieniu

Po uruchomieniu jednostki sterującej transmisją synchroniczną fizyczny kanał łączności będzie mógł realizować transmisję synchroniczną z prędkością 2400 bodów.

- ii/Liczba fizycznych kanałów łączności może się dynamicznie zmieniać podczas pracy systemu bez potrzeby jego rekonfiguracji . Jedyną odczuwalną przez użytkowników konsekwencją tych zmian mogą być zmiany czasu reakcji systemu.
- iii/System teletransmisji umożliwi użytkownikom minikomputera Mera-400 zdalny dostęp do zasobów komputera Odra-1305. Liczba użytkowników jednocześnie korzystających z systemu teletransmisji na Merze-400 jest dowolna - praktycznie ograniczona liczbą dostępnych terminali. Użytkownicy ci mają wrażenie, że działają z końcówek podsystemu MOP systemu operacyjnego GEORGE-3.
- iv/ Oprócz wszystkich możliwości podsystemu MOP użytkownicy systemu teletransmisji na Merze-400 mogą ze swoich terminali inicjować transmisje zbiorów znakowych pomiędzy systemami CROK-4 i GEORGE-3 w obu kierunkach, Transmisje te wykonują się współbieżnie z pracą użytkownika, ten, nie musi on czekać na zakończenie przesyłania zbioru aby kontynuować sesję MOP-u.
- v/ Liczba użytkowników jednocześnie korzystających z systemu teletransmisji i liczba jednocześnie przesyłanych zbiorów jest niezależna od liczby fizycznych kanałów łączności pomiędzy maszynami Mera-400 i Odra-1305. W skrajnym przypadku kilku użytkowników i kilka transmitowanych zbiorów może korzystać z jednego fizycznego kanału łączności lub jeden użytkownik może korzystać

z kilku fizycznych kanałów łączności.

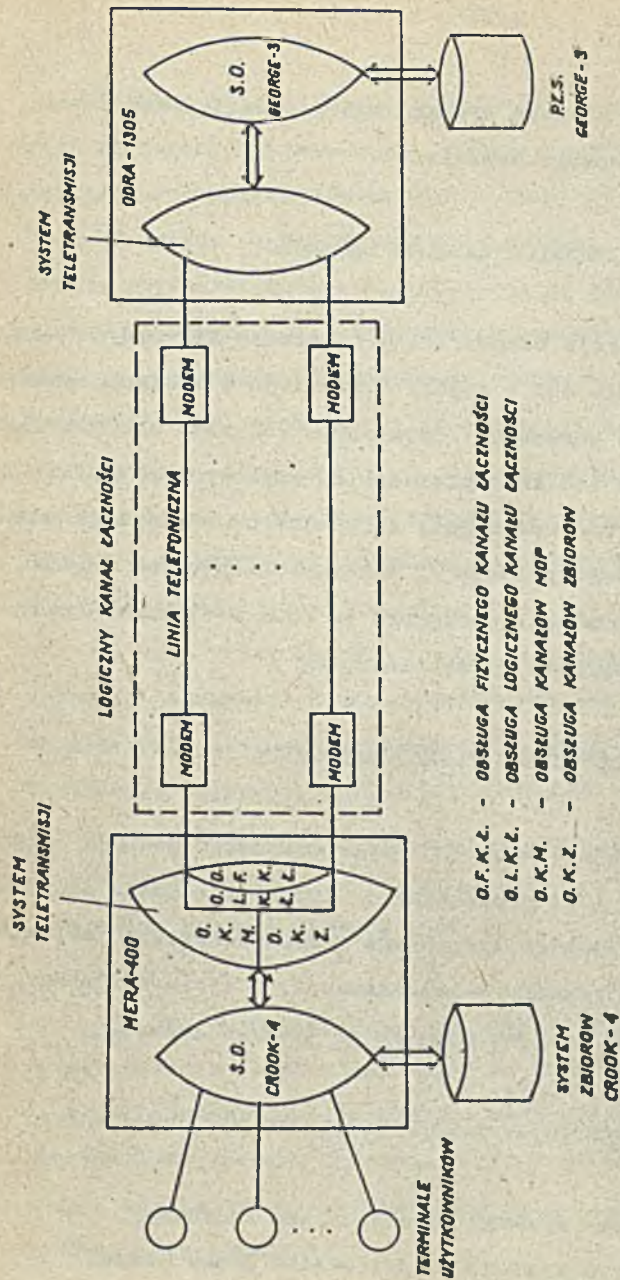
- vi/ Minikomputer Mera-400 na którym wykorzystywany jest system teletransmisji zachowuje wszystkie cechy wielodostępnego systemu cyfrowego - z terminali nie wykorzystywanych w danej chwili do współpracy z systemem teletransmisji dostępne są wszystkie możliwości oferowane przez system operacyjny CROBK-4. System teletransmisji nie jest związany na stałe z żadnymi terminalami - jest on dostępny w dowolnym momencie z dowolnego terminala.
- vii/Oprogramowanie systemu teletransmisji wykrywa i eliminuje błędy przesyłania /przekłamania w fizycznych kanałach łączności/, reaguje na awarie systemów operacyjnych i zapewnia synchronizację po ich restartach.

2. OPROGRAMOWANIE SYSTEMU TELETRANSMISJI

Oprogramowanie systemu teletransmisji rezydujące w minikomputerze Mera-400 i komputerze Odra-1305 ma budowę hierarchiczną, składającą się z trzech warstw. Warstwy te to:

- obsługa fizycznych kanałów łączności,
- obsługa logicznego kanału łączności,
- obsługa kanałów MOP i kanałów zbiorów.

Najniżej w hierarchii znajduje się warstwa obsługi fizycznych kanałów łączności. Bezpośrednio pod nią położona jest warstwa obsługująca logiczny kanał łączności, na który składają się wszystkie fizyczne kanały łączności. Najwyższa warstwa składa się z dwóch części - obsługi kanałów MOP oraz kanałów zbiorów. Ogólna struktura systemu teletransmisji scharakteryzowana została na rys. 1.



- O.F.K.Z. - OBSŁUGA FIZYCZNEGO KANAŁU ŁĄCZNOŚCI
- O.L.K.Z. - OBSŁUGA LOGICZNEGO KANAŁU ŁĄCZNOŚCI
- O.K.M. - OBSŁUGA KANAŁOW MOP
- O.K.Z. - OBSŁUGA KANAŁÓW ZBIORÓW

RYS. 1. STRUKTURA SYSTEMU TELETRANSMISJI
POMIĘDY MASZYNAMI MERA-400 I ODRA-1305.

Poniżej omówione zostaną krótko funkcje realizowane przez poszczególne warstwy oprogramowania.

3. WARSTWA OBSŁUGI FIZYCZNYCH KANAŁÓW ŁĄCZNOŚCI

Warstwa ta nadzoruje bezpośrednio działanie fizycznych kanałów łączności. Inicjuje ona i kończy transmisje w liniach łączności pomiędzy maszynami Mera-400 i Odra-1305. Cdebrane z linii komunikaty przekazuje bez ich interpretacji do warstwy obsługi logicznego kanału łączności, zaś komunikaty otrzymane od logicznego kanału łączności nadaje do współpracującej maszyny. Ewentualna zmiana fizycznych środków łączności pociągnie za sobą konieczność modyfikacji jedynie tej warstwy oprogramowania.

4. WARSTWA OBSŁUGI LOGICZNEGO KANAŁU ŁĄCZNOŚCI

Warstwa ta realizuje logiczny kanał łączności pomiędzy maszynami Mera-400 i Odra-1305 niezależny od liczby i rodzaju fizycznych kanałów łączności. Zadaniem logicznego kanału łączności jest nadzorowanie pracy całego systemu teletransmisji, a w szczególności:

- rozpoczynanie pracy systemu poprzez nawiązanie łączności ze współpracującą maszyną,
- zatrzymywanie działania systemu w przypadku chwilowej utraty łączności,
- wznowianie działania systemu po odzyskaniu łączności,
- wznowianie działania systemu po przerwaniu pracy jednej z maszyn,
- kończenie pracy systemu.

Warstwa obsługi logicznego kanału łączności sprawdza poprawność odbieranych komunikatów, żądając retransmisji komunikatów zawierających przekłamania oraz interpretuje potwierdzenia komunikatów nadanych do współpracującej maszyny, ponownie nadając negatywnie potwierdzone lub niepotwierdzone komunikaty. Kontrolowana jest również sekwencja odbieranych komunikatów, co pozwala na eliminację przypadków kilkakrotnego odebrania tego samego komunikatu. Kanał logiczny dokonuje też klasyfikacji komunikatów ze względu na adresatów, dla których są one przeznaczone. Komunikaty adresowane do kanału logicznego powodują odpowiednie zmiany stanu tego kanału, zaś komunikaty adresowane do wyższych warstw oprogramowania przekazywane są do miejsc przeznaczenia.

Opisany wyżej sposób działania logicznego kanału łączności zapewnia poprawność i niezawodność wymiany komunikatów pomiędzy najwyższymi warstwami oprogramowania systemu teletransmisji realizującymi w maszynach Mera-400 i Odra-1305 - warstwę obsługi kanałów MOP i kanałów zbiorów.

5. WARSTWA OBSŁUGI KANAŁÓW MOP

Warstwa ta znajduje się na najwyższym poziomie hierarchii oprogramowania razem z opisaną w następnym punkcie warstwą obsługi kanałów zbiorów. Jest to warstwa widziana bezpośrednio przez użytkowników systemu teletransmisji - za jej pośrednictwem prowadzą oni ze swych terminali dialog z podsystemem MOP systemu operacyjnego GEORGE-3.

Pojedynczy kanał MOP obsługuje pracę jednego użytkownika. W systemie może istnieć jednocześnie wiele takich kanałów. Kanał

MOP poprzez logiczny kanał łączności wysyła i odbiera komunikaty do i od użytkownika pracującego przy terminalu MERY-400 oraz nadaje lub przyjmuje komunikaty do lub od systemu operacyjnego GEORGE3 maszyny ODRA-1305. Oprócz komunikatów zawierających dane kanał MOP nadaje i odbiera komunikaty sterujące jego pracą, co umożliwia np. automatyczne wstrzymanie wyprowadzania informacji po wypełnieniu ekranu monitora i wznowienie wyprowadzania na żądanie użytkownika.

6. WARSTWA OBSŁUGI KANAŁÓW ZBIORÓW

Warstwa ta rezyduje na tym samym poziomie, co warstwa obsługi kanałów MOP. Umożliwia ona przesyłanie zbiorów znakowych w obydwu kierunkach pomiędzy podsystemami zbiorów systemów operacyjnych CROOK-4 i GEORGE-3. Kanały zbiorów naktywne są w wyniku realizacji zleceń transmisji zbiorów pochodzących od użytkowników działających w kanałach MOP. Każdy kanał zbioru jest związany z tym kanałem MOP, z którego pochodziło zlecenie inicjujące transmisję danego zbioru. Kanał zbioru wysyła do związanego z nim kanału MOP informację o zakończeniu swojej działalności. Jeden kanał zbioru służy do przesyłania jednego zbioru, z tym że w systemie może istnieć jednocześnie wiele kanałów zbiorów. Kanał zbioru poprzez logiczny kanał łączności przesyła kolejne rekordy zbioru z Mery-400 do zbioru na Odrze-1305 lub odwrotnie, w zależności od zadanego kierunku transmisji. Oprócz rekordów danych przesyłane są komunikaty sterujące stanem kanału zbioru.

UWAGI KOŃCOWE

W chwili obecnej omawiany system teletransmisji znajduje się w fazie testowania. Jego wdrożenie do normalnej eksploatacji powinno nastąpić w pierwszym kwartale 86 roku.

Jacek Pulwarski

S I C O B '85

Trzydzieste szóste już międzynarodowe targi informatyki, telematyki, łączności i biurotyki SICOB'85 zgrupowały 850 wystawców i 3.500 wyrobów. Trzydzieste szóste, jeśli nie liczyć dwóch mniejszych wystaw SPECIAL-SICOB organizowanych dodatkowo, od zeszłego roku na wiosnę.

Impreza od 27 lat odbywa się w wielkiej hali CNIT w podparyskiej dzielnicy LA DEFENSE. Tyle to już lat ma ów bardzo przestronny /220.000 m² powierzchni wystawowej/ i nowoczesny na pierwszy rzut oka budynek. Pierwszy zaś SICOB odbył się w roku 1950 przy udziale 150 wystawców na powierzchni 7.000 m²...

Kierunkiem przewodnim imprezy jest oczywiście mikroinformatyka i to w dodatku kompatybilna /zarówno urządzenia jak i oprogramowanie/ ze standardami IBM-PC. Urządzeń tych jest coraz więcej a nieprofesjonalistom trudno zrozumieć, jakie między nimi mogą być różnice. Na niekompatybilność z IBM-PC mogą sobie pozwolić mocno stojące firmy jak np.: HEWLETT-PACKARD czy DECK, inne zaś- jeśli chcą wejść na rynek nasycony sprzętem i /co istotne/ oprogramowaniem muszą trzymać się standardów. Spośród wielu mikrokomputerów kompatybilnych z PC/XT można wymienić TEXAS-T.I.P.C, BULL MICRAL 30, COMMODEORE PC-10, GOUPIL G4, ERICSSON PC, OLIVETTI M24, LEANORD ELAN i.t.d. Na rynku polskim poza IBM-PC dostępnym za dewizy istnieje wiele urządzeń montowanych przez firmy polonijne i dostępnych za złotówki

/QUASAR, IMP 86 i inne/, tak więc temat ten to nie legenda o żelaznym wilku.

Zeszłoroczne dziecko IBM, potężny PC/AT stał się również standardem i coraz więcej widać mikrokomputerów z nim kompatybilnych: NCR-PC8, TEXAS BUS. PRO, ZENITH Z200 PC-AT, TELEVIDEO PC-AT, COMPAQ DESKPRO-286. Na szczególną uwagę zasługuje ostatni z wymienionych, a właściwie jego producent, amerykańska firma z Teksasu, istniejąca nieco ponad dwa lata, która osiągnęła dotychczas łączne obroty ok. 450 mln \$. /w roku 1984 196% wzrostu/. COMPAQ wykorzystywał chwilową słabość IBM, który nie mógł zrealizować wszystkich zamówień na PC/AT i wszedł na rynek z modelem DESKPRO 286, mającym lepsze parametry od IBM PC/AT: poza częstotliwością zegara 6 MHz na jakiej pracuje IBM, może działać na 8 MHz /25 - 30% szybciej/, pamięć operacyjna RAM może być rozszerzona do 8,2 Mb /5Mb dla IBM/. Wcześniej niż IBM, COMPAQ wypuścił na rynek wersję przenośną tego mikrokomputera pod nazwą PORTABLE 286 /waga 14,8 kg/.

Tak więc najczęściej spotykanym na SICOB-ie słowem była kompatybilność /z IBM/, wg przedstawicieli firm zawsze stop procentowa.

Niektóre z pokazywanych urządzeń są rzeczywiście kompatybilne ! Często jest to jednak określenie na wyrost, używane w celach komercyjnych, a mit pryska przy próbie uruchomienia oprogramowania wykonanego na IBM-PC - tak to bywa gdy domniemane kompatybilne urządzenia oparte są na innym procesorze czy też akceptują inne urządzenia peryferyjne. Badaczom kompatybilności firma EXXA oferowała za 25 \$ program -test, określający procent tejeż badanego mikrokomputera. No cóż, jedni robią business na tym, czego wstydzą się inni. Jeśli jesteśmy przy IBM-ie to warto wspomnieć nowość przez tę firmę wystawioną - mikrokomputer

IBM 36 Super Compact /5364/ kompatybilny z serią 36, wielostanowiskowy. Konfigurację z pamięcią 256 Kb /rozszerzalna do 512 kb/, z pamięcią dyskietkową 1,2 Mb, z dyskiem twardym 40 Mb, drukarką, monitorem i dodatkowo mikrokomputerem PC proponowano za 12,5 tys. \$.

Rozwój mikroinformatyki idzie w kierunku zastosowań procesorów 32-bitowych /w przyszłości 64/, stosowanych od pewnego już czasu w minikomputerach. Jedną z firm idących za tym trendem jest HEWLETT PACKARD, który wypuścił na rynek nową serię 300, opartą na procesorze Motorola MC 68010 /10 MHz/ lub MC 68020 /16 MHz/. Najprostszy model z 512 Kb pamięci oferowano za 8.000 \$, zaś konfiguracje z pamięcią 2 MB, 19-calowym kolorowym ekranem i pełną grafiką - 40.000 \$. Są to ceny dość wysokie, ale wyroby HP cieszą się opinią dopracowanych do najdrobniejszych szczegółów i wysoce niezawodnych. Inną ciekawostką HEWLETT PACKARD'a był INTEGRAL PC - kompletny systemik o wadze 11,4 kg, zawierający : pamięć RAM 512 Kb, rozszerzalną do 5 Mb, pamięć ROM 256 Kb /zawierającą system operacyjny HP-UX, Personal Application Manager i "okienka" HP/, wbudowaną drukarkę atramentową /150 znaków na sekundę/, płaski ekran uchylny /24 linie po 80 znaków lub 255 x 512 punktów/ klawiaturę i stację dyskietek $3\frac{1}{2}$ cala o pojemności 710 Kb. Przy okazji zauważyć warto, że owe dyskietki $3\frac{1}{2}$ calowe stają się standardem wypierając powoli $5\frac{1}{4}$ calowe. Wprowadzili je na rynek Japończycy, a konkretnie SONY. Mają one twardą obudowę, co zapobiega uszkodzeniom mechanicznym i pozwala na zwiększenie precyzji zapisu. Łatwo zauważyć że jedna taka dyskietka dwustronna, podwójna gęstość, ma pojemność dwóch $5\frac{1}{4}$ calowych w standardzie IBM-PC.

Kolejną ciekawostką, uznaną za "wyrób roku 1985", był wielostanowiskowy supermikrokomputer/choć należałoby napisać mini

firmy NCR pod nazwą TOWER 32, oparty oczywiście na procesorze 32 - bitowym MC 68020, pracujący w systemie operacyjnym UNIX SYSTEM V, wersja 5.2. Może on wykonywać 2 miliony operacji na sekundę, dopuszcza 32 stanowiska pracy, pamięć operacyjna może być rozszerzona do 16 Mb, zaś dyskowa do 4,6 Gb /gigabajtów/. Konfiguracja z pamięcią operacyjną 1 Mb, dyskową 44 Mb, dyskiem zabezpieczającym 45 Mb, akumulatorem awaryjnym, systemem UNIX V i Business Basic'em proponowana jest /na razie tylko w USA/ za 21.900 \$.

Entuzjastów, którzy nagle zapragnęli nabyć taki sprzęt, zakładają płaszcze i wybiegają poń do sklepu, muszą nieco zmartwić wszystkie mikrokomputery oparte na procesorze 32-bitowym objęte są, jak na razie, zakazem eksportu do Polski.

Warto zauważyć, że firma NCR, znana i u nas choćby z instalacji w NBP i PKO, coraz lepiej wypada na rynku światowym. Coraz więcej mówi się o jej produktach, którymi są nie tylko komputery w tradycyjnym znaczeniu, ale i systemy zarządzania np. wielkimi sklepami samoobsługowymi, gdzie kasy wyposażone w optyczne czytniki etykiet paskowych nie tylko rachują automatycznie należność od klienta za zakupy, ale pozwalają też prowadzić na bieżąco, całkowicie w sposób automatyczny, ewidencję rozchodów wszystkich towarów; poza tym kasjerka nie musi znać ceny ani nawet nazw sprzedawanych artykułów - wystarczy że przejedzie rodzajem pióra świetlnego po etykietkach wszystkich zakupionych przez klienta towarów - a komputer zrobi resztę. NCR produkuje też np. automatyczne okienka bankowe, które pozwalają spragnionym gotówki w środku nocy zaspokoić to straszliwe pragnienie. Oferta NCR na SICOB-ie była jedną z najbogatszych: mikrokomputery PC 6 i PC 8, kompatybilne odpowiednio z IBM

PC/XT i PC/AT, opisany wyżej TOWER 32, superminikomputer 9400, procesor komunikacyjny COMTEN 5620, bankowe okienka automatyczne 5084 i terminale samoobsługowe LSB1 dla banków i agencji ubezpieczeniowych.

Mimo tylko 10% wzrostu sprzedaży w roku 1984 /ponad 3,6 mld \$/ NCR zajmuje 5 miejsce na światowej liście gigantów Gartnera. Pierwsze miejsce to oczywiście IBI /sprzedaż w r. 1984 43,5 mld \$ przy 17% wzrostu w stosunku do 1983/, drugie - DIGITAL-EQUIPMENT czyli DEC /odpowiednio 6,2 mld \$ i 29%/, trzecie - BURROUGHS /4,5 mld \$ i 12%/ i czwarte - CONTROL DATA /3,7 mld \$ i 7%/ . Wymienione wartości sprzedaży dotyczą systemów komputerowych, peryferii, urządzeń biurowy, teleinformatyki, CAD/CAM, oprogramowania i serwisu. Lista ta obejmuje 100 firm amerykańskich; ze znanych w Polsce - HP zajmuje miejsce 7, WANG-8 APPLE-9, COMODORE-12, TEXAS INSTRUMENTS-21, DATAPOINT-27, COMPAQ-43 /skok z 90/ ITT-48. Warto parę słów napisać o nowym produkcie światowego Nr 2, czyli firmy DEC, komputerze VAX 8500 /zwanym też VENUS/, mimo że ku zdziwieniu wielu zwiedzających firma ta nie zaszczyliła SICOB-u swoim uczestnictwem. Ow supermini-komputer uznany również za "produkt roku 1985", reprezentuje tzw. architekturę VAXCLUSTER.

Pamięć zbudowana jest z kostek RAM MOS 256 kilobitowych z obwodami wykrywania i poprawy różnorodnych błędów, przy poziomie integracji do 4Mb na płycie, w konfiguracji maksymalnej-32 Mb.

Cykl maszynowy wynosi 80 ns, szybkość obliczeń - 4 mln. operacji na sekundę, pole adresacji - 4 Gigabajty, magistrała 32 bitowa.

Istnieje możliwość podłączenia doń maksymalnie 512 terminali w sieci DECNET. Wg specjalistów sprzęt ten jest poważnym krokiem naprzód - brak niestety miejsca na szersze

rozważania. Jako ciekawostkę można podać, iż konsolę operatorską stanowi komputer PDP-11 z 256 Kb pamięci i dyskiem 10 Kb, zapewniający nie tylko normalne sterowanie pracą VAX-a, ale i kontrolę jego działania /m.in. możliwość stałego automatycznego uruchamiania testów naszynowych/. Poza samymi komputerami niejaki postęp daje się zauważyć w urządzeniach peryferyjnych, szczególnie w drukarkach. Niektóre drukarki matrycowe mogą drukować w kolorze, jak np. JAPY 615, drukująca w 7 kolorach + czarny z wydajnością do 400 znaków na sekundę.

Coraz większy staje się też zakres znaków możliwych do wydrukowania /programowalne/. Szybkość pracy komputerów stale rośnie, natomiast szybkość druku ustabilizowała się na poziomie ograniczonym przez mechanikę. Przyszłość należy więc do drukarek bezuderzeniowych, przykładowo laserowych. SIEMENS proponuje laserową drukarkę model 2300/ dającą się podłączyć m.in. do IBM 370/, z wydajnością 206 stron na minutę, zaś XEROX, /model 4045/ nieco wolniejszą / 10 stron na minutę/ ale za to tańszą /8.000 \$/ i dającą się podłączyć jednocześnie do 4 komputerów. Poza laserami wielką nowością jest projekcja jonów; technikę tę zastosowała firma IER w drukarce DELPHAX 6000, mającą wydajność 60 stron A4 na minutę z precyzją 240 x 240 punktów na cal² !. Ta ostatnia wartość jest nie do osiągnięcia przy druku mechanicznym, ale za to szybkość druku może wywołać uśmiech politowania u wytwórców drukarek taśmowych /czyli "karabinów maszynowych"/. Jednakże dla tych, którzy dłużej pracowali obok tych ostatnich, względna nawet cisza nie ma ceny, a to zapewnia pełny sukces drukarkom bezuderzeniowym. Niektóre drukarki stają się prawie komputerami jak np. laserowa APPLE WRITER, oparta o procesor 68000 z pamięcią 1 Mb.

Jeśli chodzi o pamięci dyskowe, coraz więcej pojawia się dysków optycznych, mających tę wadę że można je zapisać tylko raz, ale za to posiadających ^{dużą} pojemność i olbrzymią precyzję zapisu.

Kadają się więc wspaniale do przechowywania oprogramowania systemowego, często używanego i nie zmienianego. Firma ISI proponowała dysk optyczno-numeryczny WORM o pojemności 100 Mb i średnicy $5\frac{1}{4}$ cala.

W tradycyjnej technologii warto wspomnieć dyski typu Winchester: CONTROL DATA KID-2 9272 średnicy 14 cali i pojemności 858 Mb oraz MITSUBISHI KR 522 średnicy $3\frac{1}{2}$ cala i pojemności 20 Mb z czasem dostępu 70 ms.

Dość chyba już tych wszystkich technicznych szczegółów i siódmym cudów świata, trzeba pomału wracać na ziemię i zabawać się tym co na naszym rynku dostępne. Warto też nadmienić, że z producentów z krajów socjalistycznych, dwóch wzięło udział w paryskiej ekspozycji: ROBOTRON /NRD/ i ISOTIMPEX /Bułgaria/. Wystawiane produkty ani same w sobie, ani tym bardziej w porównaniu z innymi nie były rewelacyjne i stoiska te zwiedzających raczej nie przyciągały. Ale jednak sama obecność na takim forum też się liczy. Jeśli już jesteśmy przy informatyce w krajach socjalistycznych to warto może przypomnieć, iż trwają negocjacje między francuskimi firmami THOMSON i LEONARD /które są głównymi dostawcami sprzętu komputerowego do szkół francuskich/ a Związkiem Radzieckim w sprawie dostarczenia naszemu sąsiadowi ok. 10.000 tzw. nanosieci składających się z komputerów T07/70 i M05. Inne negocjacje dotyczą informatyzacji kołchozów i tu znów chodzi o sprzęt francuski-mikrokomputery COUPEL firmy SET. W najbliższym czasie Francja wybuduje w Chinach wielkie centrum badań nad mikroelektroniką i dostarczy 100 000 linii telefonicznych /firmy ALCATEL-THOMSON/ dla miasta

Pekin.

U nas zaś dobry interes, jak już wcześniej wspomniałem, robią firmy polonijne, sprowadzające komponenty i montujące je w mikrokomputery. Dobry interes nie tylko dla siebie, ale i dla polskiej informatyki, jako że urządzenia te można nabyć szybko /bywa że w tydzień po złożeniu zamówienia/, tanio /tzn mniej więcej za tyle, za ile na rynku państwowym można nabyć bardzo przestarzałe już modele mikrokomputerów/ i za złotówki. Niektórych bulwersują duże zyski tych firm /podobno 1000-2000 zł za 1 dolara/, ale chwala polonusom za to, że umożliwili nam szybki dostęp do w miarę nowoczesnej światowej mikroinformatyki, bez której trudno dziś sobie wyobrazić prawidłowy rozwój gospodarki.

II OPROGRAMOWANIE SYSTEMOWE

Metodologia programowania w LOGLANie

A. Salwicki
Instytut Informatyki
Uniwersytet Warszawski
PKiN p.850
00-901 Warszawa

STRESZCZENIE

Celem tego artykułu jest wykazanie że dwa narzędzia "teoretyczne" -specyfikacja oprogramowania w terminach logiki algorytmicznej i "praktyczne" - język programowania LOGLAN uzupełniają się nawzajem i tworzą całość o walorach użytkowych niebagatelnych, które trudno dziś w pełni ocenić.

WSTĘP

Znaczenie oprogramowania stale rośnie. Najlepiej świadczy o tym stale rosnący udział oprogramowania w obrotach na światowym rynku informatycznym. Przekroczył on już 80% całości obrotów i nie zanosz się na to by miał zmaleć w przyszłości, wprost przeciwnie, stale rosną: rozmiary oprogramowania, obszar różnorodnych zastosowań komputerów i rozmiar zadań jakie rozwiązujemy posługując się komputerami. Szybkość oferowania nowych systemów ma olbrzymie znaczenie handlowe i technologiczne. Postęp gospodarczy nie może być osiągnięty bez informatyzacji gospodarki. Trudno sobie wyobrazić by bez komputeryzacji udało się zwiększyć wydajność, zmniejszyć zużycie surowców, kto zresztą zechce kupować produkty nie zawierające mikroprocesorów? Z drugiej jednak strony rośnie koszt błędów popełnianych podczas tworzenia oprogramowania, zarówno bezpośredni koszt ich usuwania z systemów jak i efekt opóźnień w stosunku do firm konkurencyjnych, opóźnień we wprowadzaniu nowej produkcji etc.

Potrzebne są pewne i wydajne metody produkcji oprogramowania. Przeszło ćwierć wieku temu powstały pierwsze kompilatory. Usprawniły one i przyspieszyły prace nad programowaniem komputerów w bardzo istotny i znaczący sposób. Program napisany w języku wysokiego poziomu pozwala skupić się na istocie zadania jakiego algorytm ma rozwiązać i uwalnia od troski o detale. Rychło okazało się jednak, że istnieją granice przydatności tych narzędzi. Nadawały się one do obliczeń numerycznych, inżynierskich. Tam jednak gdzie przedmiotem obliczeń miały być obiekty, które trzeba było kodować w pewien sposób w pamięci, tam języki typu ALGOL czy FORTRAN niewiele pomagają. Próbowano temu zaradzić wprowadzając nowe typy danych np. liczby zespolone do FORTRANu czy rekordy i wskaźniki do PASCALA. Podejściom tym brakowało jednak konsekwencji i prostoty stowarzyszonej z uniwersalnością.

Kierunki jakie wyraźnie się rysują jako przyszłe standardy w pracy nad tworzeniem oprogramowania to:

- stosowanie komputera w coraz obszerniejszym zakresie prac programisty, a więc nie tylko edytory i kompilatory ale i systemy wspomagające wcześniejsze fazy pracy. Wydaje się że efektem (jednym z wielu) będzie odejście od pojmowania programu jako pewnej całości, jako tekstu zbudowanego z linii. Wzmoczona potrzeba korzystania z bibliotek, procedur, modułów dostarczanych przez innych wskazuje na potrzebę pojmowania programu jako struktury zbudowanej z modułów. W programach powinny też występować wyrażenia, które nie mają znaczenia dla wykonania programu - specyfikacje. Pewne kroki w tym kierunku już poczynili ci którzy wpisują w formie komentarzy czy też asercji warunki jakie mają być spełnione gdy program osiągnie wskazany punkt sterowania. Potrzebne są jednak jeszcze inne formy bądź cele specyfikowania programów: w dokumentacji powinny być zanotowane już szkice koncepcji, nie tylko gotowe algorytmy, powinny też wystąpić specyfikacje abstrakcyjnych typów danych.

Ale co to jest abstrakcyjny typ danych bądź abstrakcyjna struktura danych? Wielokrotnie stwierdzamy, że postawione nam zadanie powinno być sformułowane w języku operacji i relacji

jakich nie dostarcza nam język programowania jakim dysponujemy. Można temu zaradzić wprowadzając specjalne, problemowo-zorientowane języki programowania: dla działań na zbiorach, dla działań na obiektach graficznych, dla księgowości, projektowania maszyn, układów elektronicznych etc. Nie widać końca wyliczeniu z poprzedniego zdania. Skłania to do przyjęcia następującej dyrektywy: dostarczyć sposobów dla określania cech jakie powinny posiadać potrzebne nam narzędzia: operacje i relacje a także dostarczyć sposobów implementacji tych operacji. Kolejnym punktem naszego sposobu pojmowania pracy nad oprogramowaniem jest konieczność szerszego posługiwania się metodami formalnymi. Wynika to z konieczności zapewnienia lepszej efektywności pracy zespołów programistów, którzy muszą przecież współdziałać, komunikować się między sobą. Nie sposób zwiększyć wydajności działania bez przekazywania specyfikacji, dyskusji nad nią, jej ulepszania. Obecnie odbywa się to w języku naturalnym, etnicznym wzbogacanym żargonem jaki wytwarza się w zespołach. Ten sposób sprzyja powstawaniu subiektywnych rozumień specyfikacji, całkowicie uniemożliwia zwracanie uwagi zespołu na kwestie tak istotne jak niesprzeczność specyfikacji, jej pełność etc. Dalej, należy oczekiwać, że języki jakie powstają będą językami dla tworzenia systemów nie tylko dla tworzenia algorytmów. Takie języki już powstają i będą się doskonalić. Przewidywać też należy lepsze, powszechniejsze wykorzystywanie wiedzy o złożoności zadań i o algorytmach oraz strukturach danych umożliwiającą większą efektywność obliczeń. Idzie tu nie tylko o obniżenie kosztów obliczeń ale i o zwiększenie rozmiaru zadań rozwiązywanych na danym komputerze.

Abstrakcyjne typy danych

Algorytm operuje na danych z pewnego zbioru wykonując na nich operacje i sprawdzając czy zachodzą relacje pomiędzy nimi. W ten sposób pojawia się pojęcie systemu algebraicznego lub inaczej struktury danych.

Strukturą danych jest układ :

- zbiór,
- operacje na elementach tego zbioru,
- relacje pomiędzy elementami tego zbioru.

Przykłady

<zbiór liczb naturalnych, dodawanie, mnożenie, równość>#

Analizując budowę komputera można dopatrzeć się struktury danych zwanej arytmetyką komputerową np.

<zbiór słów binarnych 16-bitowych;
operacje na słowach np. dodawanie, przesunięcia,
relacje m.in. równość słów, być liczbą dodatnią >#

W podręcznikach języków programowania spotykamy po kilka struktur danych zwanych też typami: całkowity, rzeczywisty, boolowski, znakowy, tekstowy itp.

Podczas swojej pracy programista projektuje często algorytmy działające w strukturze danych innej od dostarczonych przez sprzęt i oprogramowanie np. stosy, kolejki, zbiory elementów ustalonego typu i in.

W r. 1974 C.A.R. Hoare zauważył, że należy wtedy zadanie podzielić na dwa podzadania:

- i) specyfikacja i implementacja struktury danych,
- ii) projektowanie, analiza i uruchamianie programu.

Zgodnie z tą sugestią podczas tworzenia programu możemy i powinniśmy abstrahować od szczegółów implementacji, natomiast powinniśmy wykorzystywać te tylko własności struktury danych, które zostały wymienione w jej specyfikacji lub mogą być z niej wnioskowane.

Na końcowy produkt programistyczny składają się dwa moduły

Moduł	Abstrakcyjny
implementujący	program
strukturę danych	

Zasada Hoare'a umożliwia wykonywanie abstrakcyjnego programu w towarzystwie różnych modułów implementujących. Jeżeli przestrzegaliśmy tej zasady to zmiana modułu implementacyjnego nie wymaga żadnych dodatkowych zmian w abstrakcyjnym programie, jego poprawność zostanie zachowana. Natomiast możemy zyskać (bądź stracić) na koszcie wykonania zadania, jeżeli lepiej (lub gorzej) dobierzemy implementację struktury danych. Podział pracy według tej zasady ma i drugą zaletę: jeden moduł implementujący może być wykorzystany przez wiele programów. Moduł taki jest więc implementacją pewnego języka problemowo-zorientowanego.

Praca nad wyżej wymienionymi modułami powinna przebiegać w kilku etapach:

- i) stworzenie specyfikacji tzn. aksjomatycznego opisu struktury danych i upewnienie się, że specyfikacja jest niesprzeczna oraz, że opisuje ona tę strukturę danych, o którą nam chodziło.
- ii) zaprojektowanie algorytmu i jego weryfikacja w oparciu o specyfikację.
- iii) zrealizowanie struktury danych (implementacja) i weryfikacja jej poprawności tzn. zgodności ze specyfikacją.

Symulacja

Obecnie omówimy zadanie skonstruowania modułu symulującego oddział banku. Zadanie nasze możemy rozwiązać stosując kilkakrotnie zasadę podziału Hoare'a. A więc najpierw opiszemy pojęcia potrzebne dla przeprowadzenia symulacji: są to klient banku, urzędnik banku, okienko. W tych terminach można już opisać działanie oddziału banku np. rozwijając poniższy szkic: klienci wchodzą do banku wg pewnego pseudolosowego procesu, podejmują decyzję jaka operacja bankowa ma być wykonana, zajmują pozycję w kolejce do odpowiedniego okienka i są obsługiwani przez urzędnika banku zgodnie z kolejnością zgłoszenia się. Sformalizowanie tego algorytmu w postaci abstrakcyjnego algorytmu nie przedstawia większych trudności jeżeli możemy posłużyć się operacjami z abstrakcyjnej struktury danych oddział bankowy np. kreowanie obiektu klient banku, zajęcie pozycji w kolejce do okienka, rozpoczęcie obsługi przez urzędnika etc. Mamy więc nas-

tępujący obraz

Moduł oddziału	Program symulacji
banku	zachowania oddziału

Z kolei, gdy rozważamy sposób realizacji oddziału banku widzimy że jest to struktura danych w której pojawiają się obiekty dwu typów: klient i obsługa, że klienci są tworzeni (dokładniej ma być określona operacja kreowania klienta) i że mogą stawać w kolejkach. Nasuwa się propozycja zrealizowania modułu oddziału banku poprzez zdefiniowanie zestawu programów abstrakcyjnych posługujących się pojęciami z modułu bardziej ogólnego: biuro.

Biuro	Oddział Banku	Program symulacji
-------	---------------	-------------------

Dalsza analiza wskazuje że potrzebne są nam pojęcia jeszcze bardziej ogólne: proces symulowany i odpowiednie operacje związane z tym pojęciem, pojęcie zdarzenia w symulowanym systemie etc. Gdy rozważymy jak rozumieć pojęcie zdarzenia okaże się iż potrzebujemy kolejnego aparatu pojęciowego: struktury danych kolejka priorytetowa, w kolejce priorytetowej zapisujemy informacje o zdarzeniach. Informacje te składają się z nazwy procesu i czasu w jakim wystąpi zdarzenie (jest to z reguły zdarzenie polegające na wznowieniu wskazanego procesu). Należy więc zbudować strukturę danych kolejka priorytetowa. Acha! zapomnieliśmy o kolejkach w jakich mają stawać klienci. Wszystko razem wydaje się mocno skomplikowane i zupełnie niejasne. Za chwilę rozplączemy ten węzełek, przedtem jednak zanotujmy jakie moduły mogą nam być przydatne.

Kolejki	kolejki priorytetowe	moduł symulacji
		ogólny

Biuro	oddział banku	program symulacyjny
-------	---------------	---------------------

A więc razem sześć modułów, pięć z nich to moduły implementujące struktury danych a tylko jeden jest programem (abstrakcyjnym - bo wykorzystującym pojęcia i operacje określone w innych modułach). Co ma być treścią modułu implementującego strukturę danych? Ależ oczywiście definicje typów i deklaracje procedur realizujących operacje właściwe dla kolejnej struktury. Jakie własności mają mieć te operacje i jak je zrealizować? Byłoby błędem rozpoczynać prace nad modułem od napisania procedur, przecież nie wiemy nawet czym mają się różnić struktury kolejki i kolejki priorytetowej.

O aksjomatycznej specyfikacji pojęcia kolejek.

Czym jest system relacyjny kolejek? Większość czytelników odpowie że kolejka to skończony ciąg elementów, że do tego ciągu można wstawiać elementy na końcu i wyjmować elementy z początku ciągu. Spróbujemy te postulaty zapisać w sposób pełniejszy nie tracąc jednak na czytelności zbyt wiele.

Uniwersum systemu kolejek składa się z dwu rozłącznych zbiorów zwanych także typami:

- z podzbioru E elementów wstawianych do kolejek i
- z podzbioru Q kolejek.

Struktura kolejek nie jest zależna od natury elementów ze zbioru E. W naszym opisie zbiór E jest więc parametrem, który może być różnie pojmowany i realizowany.

Rozpatrywać będziemy następujące operacje na kolejkach:

w (wstaw) w: $E \times Q \rightarrow Q$

u (usuń) u: $Q \rightarrow Q$

p (pierwszy) p: $Q \rightarrow E$

a także relacje

n? (nie pusty?) n? : $Q \rightarrow \{\text{prawda, fałsz}\}$

= (równość) .

Struktura o takim zestawie operacji i relacji, mówimy też struktura o takiej sygnaturze,

$\langle Q \ E ; w, u, p, n?, = \rangle$

jest kolejką jeżeli spełnia następujące postulaty:

- (e) (q) $n?(w(e,q))$ co czytamy natepująco, dla każdego elementu $e \in E$ i dla każdej kolejki q zachodzi $n?(w(e,q))$
- (e) (q) $-n?(q) \Rightarrow p(w(e,q)) = e$
- (e) (q) $n?(q) \Rightarrow p(w(e,q)) = p(q)$
- (e) (q) $-n?(q) \Rightarrow u(w(e,q)) = q$
- (e) (q) $n?(q) \Rightarrow u(w(e,q)) = w(e,u(q))$
- (q) while $n?(q)$ do $q := u(q)$ od true

Ostatni postulat jest własnością algorytmiczną, jego sens semantyczny to stwierdzenie że każda kolejke można opróżnić w skończonym czasie, że jest ona skończona. Wyrażenia postaci program po którym wytepuje wyrażenie boole'owskie jest formułą algorytmiczna. Semantykę takich formuł określamy następująco: formuła algorytmiczna postaci $\langle \text{program } K \rangle \langle \text{formuła } \alpha \rangle$ jest w danym stanie prawdziwa wtedy i tylko wtedy gdy obliczenie programu K rozpoczęte w tym stanie jest skończone i gdy końcowy stan spełnia warunek α . [AL]

Potrąfimy udowodnić następujące twierdzenie (o reprezentacji)
Każda struktura danych o odpowiedniej sygnaturze

$\langle A_1, A_2; f_1, f_2, f_3, r, = \rangle$

- $f_1: A_1 \times A_2 \rightarrow A_2$
- $f_2: A_2 \rightarrow A_2$
- $f_3: A_2 \rightarrow A_1$
- $r: A_2 \rightarrow \{\text{prawda, fałsz}\}$

spełniająca powyższe postulaty jest izomorficzna ze strukturą skończonych ciągów elementów ze zbioru A_1 , tzn. ze struktura

$\langle A_1 \cup A_1^* ; d, o, p, z, = \rangle$

gdzie operacja $d(a,b)$ oznacza dopisanie elementu a na końcu skończonego ciągu b , operacja $o(b)$ oznacza ciąg powstający z ciągu b przez odrzucenie jego pierwszego elementu,

operacja $p(b)$ dostarcza pierwszy element ciągu b ,
ciąg b należy do relacji z wtedy i tylko wtedy gdy jest pusty.

Jeżeli pominiemy ostatni, algorytmiczny aksjomat stwierdzający, że program $\text{while } \neg n?(q) \text{ do } q := u(q)$ od nie zapętli się, to pozostałe postulaty mogą być spełnione przez strukturę złożoną z ciągów skończonych i pewnych ciągów nieskończonych przy odpowiedniej interpretacji działań i relacji. Wskazuje to na istotną rolę jaką odgrywa ten algorytmiczny aksjomat w opisie pojęcia kolejki. Jak się dalej okaże ta własność jest bardzo przydatna podczas weryfikacji programów abstrakcyjnych. Uzyskaliśmy specyfikację modułu kolejki. Jego struktura powinna uwzględniać następujące elementy:

KOLEJKI

typ element	
typ kolejka	
operacja wstaw	
usuń	
pierwszy	
relacja niepusty	
rownosc	

Następnym modulem, który powinien zostać opisany jest moduł kolejek priorytetowych. System kolejek priorytetowych umożliwia operacje na skończonych podzbiorach pewnego zbioru E elementów uporządkowanego przez pewną relację porządku, tu oznaczymy ją symbolem \langle . Rozważać będziemy operacje: wstaw element e do zbioru s , usuń element e ze zbioru s , sprawdź czy element $e \in E$ jest aktualnie elementem zbioru s ? wskaż element najmniejszy w zbiorze s , odpowiedz na pytanie czy zbiór s jest niepusty? Aksjomatyczna definicja klasy systemów kolejek priorytetowych istnieje, składają się na nią następujące elementy:

sygnatura

$\langle E \ S; i, d, \text{min}, \text{mb}, \text{em}, \langle, = \rangle$

$i: E \times S \rightarrow S$	$em: S \rightarrow \{\text{prawda, fałsz}\}$
$d: E \times S \rightarrow S$	$\langle: S \times S \rightarrow \{\text{prawda, fałsz}\}$
$\text{min}: S \rightarrow E$	$=: S \times S \rightarrow \{\text{prawda, fałsz}\}$
	$\text{mb}: E \times S \rightarrow \{\text{prawda, fałsz}\}$

aksjomaty:

- A1) zbiór E jest liniowo uporządkowany przez relacje \langle
- (e) $\{e \langle e\}$
- (e, e') $\{(e \langle e' \Rightarrow (e' \langle e \Rightarrow e = e'))\}$
- (e, e', e'') $\{e \langle e' \wedge e' \langle e'' \Rightarrow e \langle e''\}$
- (e, e') $\{e \langle e' \vee e' \langle e\}$
- A2) $\neg em(s) \Rightarrow [(e) mb(e, s) \Rightarrow \text{min}(s) \langle e]$
- A3) $\{s' := i(e, s)\} \{mb(e, s) \wedge e' \neq e \Rightarrow mb(e', s) \langle \Rightarrow mb(e', s')\}$
- A4) $\{s' := d(e, s)\} \{\neg mb(e, s) \wedge e' \neq e \Rightarrow mb(e', s) \langle \Rightarrow mb(e', s')\}$
- A5) while $\neg em(s)$ do $s := d(\text{min}(s), s)$ od true
- A6) $mb(e, s) \langle \Rightarrow$ begin $s1 := s$; $\text{bool} := \text{false}$;
- while $\neg em(s1) \wedge \neg \text{bool}$ do
- $e1 := \text{min}(s1)$;
- $\text{bool} := (e1 = e)$;
- $s1 := d(e1, s1)$
- od
- end bool

Zbiór aksjomatów A1 - A6 jest niesprzeczny i podobnie jak dla systemu kolejek można udowodnić twierdzenie o reprezentacji orzekające, że każdy model tego układu aksjomatów jest równoważny systemowi skończonych podzbiorów zbioru E . Określiśmy strukturę drugiego z kolei modułu.

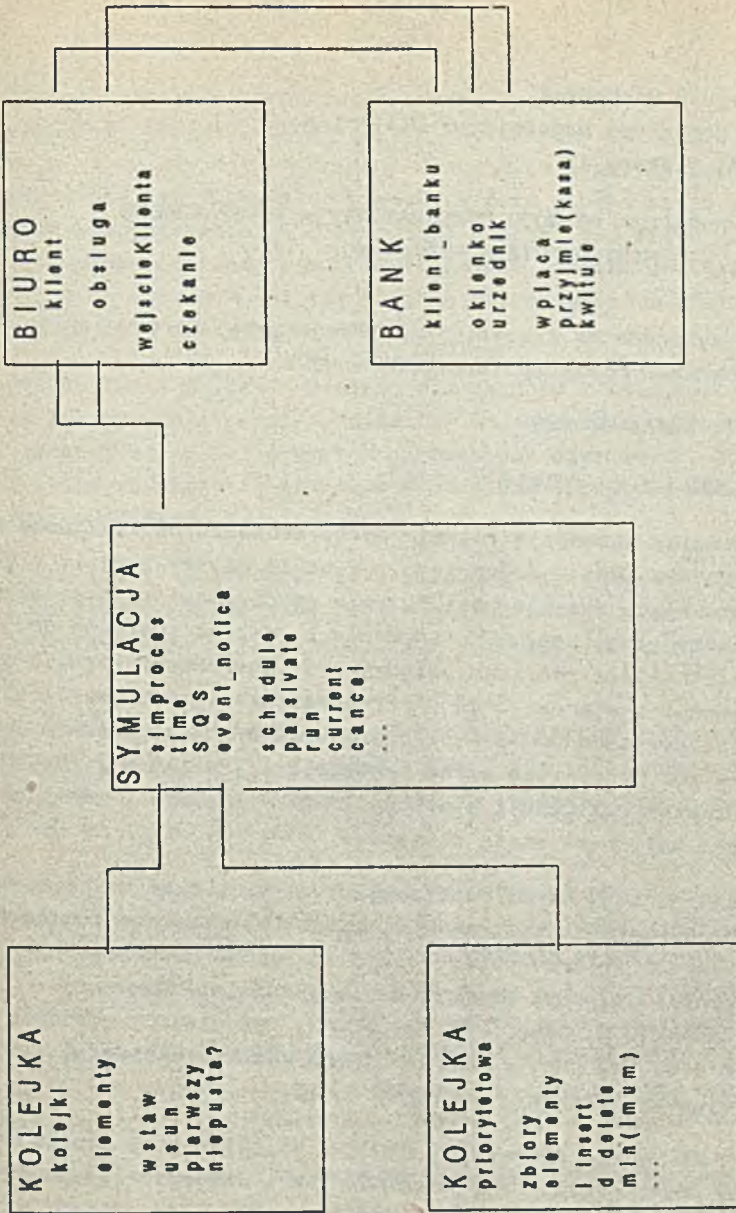
Moduł symulacji (ogólne narzędzia) powinien dostarczyć pojęć takich jak proces symulowany, operacje uaktywnienia procesu, wstrzymania, pasywacji etc. Na czym ma polegać wykonanie takiej operacji? Wyobraźmy sobie że każdy proces przechodzi od zdarzenia do następnego zdarzenia w jego scenariuszu wykonując pewne instrukcje. Zdarzenie - to chwila jego powtórnej reaktywacji.

Zdarzenia mogą więc być reprezentowane przez pary złożone z nazwy procesu i czasu zdarzenia (oczywiście idzie tu o czas symulowany). W celu planowania przebiegu procesu symulacji użyjemy kolejki priorytetowej - będzie ona reprezentować oś czasu. W kolejce priorytetowej o nazwie SQS będą umieszczane informacje o zdarzeniach tzn. pary złożone z nazwy procesu i wartości czasu t dla której proces ma być wznowiony. Zawieszanie i wznowianie procesów jest możliwe dzięki wykorzystaniu mechanizmu współprogramów. Każdy symulowany proces jest współprogramem. (Angielskie słowo oznaczające współprogram - to coroutine). System współprogramów jest zrealizowany w LOGLANie jako jego integralna część. Czytelnika zainteresowanego własnościami współprogramów odsyłamy do publikacji o LOGLANie. Moduł symulacji zawiera, oprócz innych, dwa zasadnicze pojęcia procesu symulowanego - simprocesu i informacji o zdarzeniu - eventnotice. Operacje dla nas najważniejsze to: schedule - zaplanuj wznowienie procesu o nazwie p na czas (symulowany) t , hold - wstrzymaj proces bieżąco wykonywany na t jednostek czasu, passivate - usuń proces (bieżący) z osi czasu. Moduł symulacji ma umożliwić tworzenie konkretnych procesów, między innymi ustawianie ich w kolejkach np. procesy klienci będą ustawiać się w kolejkach do procesów urzednik lub okienko. A więc pojęcie procesu symulowanego stanowi rozwinięcie pojęcia bardziej ogólnego i uniwersalnego jakim jest pojęcie element kolejki. W podobny sposób należy rozpatrzyć pojęcie informacji o zdarzeniu, taka informacja - wiemy że składa się ona z dwu składników - będzie wstawiana, odszukiwana i usuwana z osi czasu tzn. z pewnej kolejki priorytetowej. Widzimy więc że nasze zadanie polega nie tylko na połączeniu modułów: kolejka, kolejka priorytetowa i symulacja ale także na połączeniu podmodułów.

Rysunek pokazuje to nam wielokrotnie. Dla rozwiązania zadania symulacji oddziału bankowego powinniśmy połączyć moduły, połączenie takie umożliwia wykorzystanie pojęć (typów) określonych wcześniej i operacji - dokładniej, użykujemy możliwości rozwinięcia, rozszerzenia pojęć określonych we wcześniejszym module. Proces symulowany simproces jest szczególnym przypadkiem pojęcia element kolejki, klient_banku jest szczególnym przypad-

kiem pojęcia klient itd. Taka hierarchiczna metoda konstrukcji oprogramowania jest nie tylko wygodna, jest także ekonomiczna. Oto możliwe staje się składanie modułów i rozwijanie ich hierarchii, stają się one przedmiotem wymiany, handlu. Oczywiście można moduły które tu omawiamy tworzyć w PASCALu nie ma jednak w PASCALu mechanizmu składania i rozwijania modułów. Jeżeli w jednym programie stworzymy moduły tu pokazane to nie będziemy mogli wykorzystać ich w otoczeniu innego programu. Język programowania LOGLAN umożliwia pracę w naturalny sposób odzwierciedlającą rozwój pojęć w systemach hierarchicznych. Z drugiej strony ten sposób pracy dobrze zgadza się z wymogami poprawności i precyzji zmierzającej do wyleiminowania jakże kosztownych błędów.

W tym miejscu przerywamy analizę zadania symulacji oddziały bankowego. Czytelnik zechce spróbować samodzielnej specyfikacji abstrakcyjnych typów danych które odpowiadają pozostałym modułom. Implementacja wszystkich modułów znajduje się w materiałach szkoły letniej LOGLANu, Zaborów 1983 i może być uzyskana w Instytucie Informatyki Uniwersytetu Warszawskiego. Nie poruszyliśmy tu innych bardzo interesujących tematów. Zainteresowanych odsyłamy do wcale już obszernej literatury przedmiotu. Pragniemy jeszcze raz podkreślić że przedstawiona tu analiza zadania symulacji nie jest dla nas celem samym w sobie, użyliśmy jej dla przedstawienia ogólnych uwag i metod tworzenia i weryfikacji oprogramowania. Nie poruszyliśmy tu jeszcze wielu kwestii tak ważnych jak: posługiwanie się specyfikacją podczas analizy programu, implementacja struktur danych i zgodność takiej implementacji z podaną wcześniej specyfikacją. Nie omówiliśmy też konstrukcji programotwórczych jakie oferuje język LOGLAN: współprogramów, procesów współbieżnych, modułów obsługi wyjątków, dynamicznych tablic etc. Najważniejszą cechą LOGLANu jest jednak możliwość składania modułów. Jest to pojęcie zapewne nowe dla większości czytelników. Może ono wydawać się trudne ponieważ jest całkowicie odmienne od tego, z czym się dotąd w programowaniu stykaliśmy. Autor tej pracy uzna że osiągnął swój cel jeżeli zainteresował czytelników nowymi możliwościami jakie to pojęcie oferuje dla praktyki tworzenia software'u.



inż. Bogdan Machowiak
Biuro Projektów Budownictwa Wiejskiego
Poznań, Piekary 17

KONCEPCJA ROZWOJU OPROGRAMOWANIA PODSTAWOWEGO
MINIKOMPUTERA MERA - 400

Praca wykonana na zlecenie Rady Programowej Porozumienia
Użytkowników Minikomputera MERA - 400.

Poznań, kwiecień 1985 r.

1. CROOK-4 i co dalej?

Opracowując koncepcję rozwoju oprogramowania podstawowego minikomputera MERA - 400 przede wszystkim należy zwrócić uwagę na istniejące dwa kierunki rozwoju oprogramowania systemowego tej maszyny: systemy SOM-3 podobne i rodzina systemów CROOK. Najistotniejsza różnica pomiędzy tymi kierunkami rozwoju oprogramowania systemowego jest okres realizacji tych opracowań: rodzina systemów SOM-3 wyposażona była przez producenta MERY - 400 przez cały okres produkcji tej maszyny / od 1976 r./, natomiast w realizacji systemów CROOK - dopiero CROOK-4 od roku 1983 znajdował wielu nabywców.

Z uwagi na inny okres realizacji tych dwóch kierunków rozwoju oprogramowania systemowego, zasoby narzędziowe ich są również bardzo zróżnicowane. Pomijając jakość niektórych pozycji dostępnymi językami programowania w rodzinie SOM-3 są: FORTRAN-SIMBOL, PASCAL, BASIC, LOGLAN. Natomiast w systemie CROOK-4 do tej pory /kwiecień 1985/ można programować tylko następującymi językami: FORTRAN, BASIC.

Mimo, że system operacyjny CROOK-4 został bardzo wysoko oceniony zarówno przez użytkowników jak i środowiskowe grupy projektantów oprogramowania podstawowego /3/, /4/ posiada on jednak, traktując go jako zjawisko, pewne niedoskonałości.

Proces powstawania rodziny systemów operacyjnych CROOK /CROOK-2, CROOK-3, CROOK-4/ był realizowany bez szczególnego planu czy harmonogramu prowadzonych prac. Bodźcem do pracy nad tym doskonałym produktem była wielka pasja i duże zaangażowanie. I dlatego też produkt ten przybrał tak wysoko oceniane przez użytkowników parametry eksploatacyjne oraz wywołał tak duże zainteresowanie w krajowej informatyce sprzętem MERA-400.

Należy tutaj wyrazić uznanie dla zespołu z Instytutu Okrętowego Politechniki Gdańskiej za wykonanie systemu operacyjnego o tak wysokich parametrach użytkowych, co pozwoliło radykalnie polepszyć środowisko sprzętowe minikomputera MERA-400.

Można tutaj stwierdzić, że środowisko sprzętowe MERA-400 było i jest największym polem doświadczalnym w tworzeniu oprogramowania systemowego i podstawowego w kraju.

W tym samym czasie w wielu ośrodkach: Uniwersytet Warszawski, Uniwersytet Jagielloński, Instytut Ciężkiej Syntezy Organicznej i inne, powstało znaczące oprogramowanie systemowe, które znalazło zastosowanie i uznanie wśród użytkowników MERA-400.

Jednakże sukces CROOK'a-4 jest najbardziej wyraźny, przede wszystkim z następujących powodów: ideowej zbieżności z systemami operacyjnymi dla minikomputerów w światowej informatyce, bardzo dużej uniwersalności - system otwarty na dobudowanie własnych zdefiniowanych przez użytkownika zastosowań, doskonale realizowanego wielodostępu i jednocześnie pracy wieloprogramowej oraz bardzo dobrego wykorzystania sprzętu minikomputera MERA - 400.

Jednakże, traktując CROOK'a-4 globalnie jako nowe jakościowe narzędzie dla użytkowników minikomputera MERA-400, powinien posiadać ten system pełny zestaw oprogramowania podstawowego łącznie z dokumentacją.

Zestaw tego oprogramowania podstawowego: języki programowania i ich kompilatory, edytory i inne oprogramowanie podstawowe specjalistyczne nie powinno być ilościowo i jakościowo gorsze od zasobów systemu wypieranego SOM-3 czy nowych systemów SOM-3 podobnych.

Użytkownik doskonałego systemu operacyjnego CROOK-4 w chwili obecnej ma do dyspozycji tylko dwa języki programowania wysokiego poziomu BASIC i FORTRAN oraz mało uniwersalny translator języka symbolicznego ASSEMBLER. To niewielkie ilościowo /ale dobre jakościowo/ oprogramowania podstawowe CROOK'a-4 powoduje pewną przepaść pomiędzy doskonałym systemem operacyjnym a możliwościami korzystania z jego zasobów z punktu widzenia zastosowań.

Dużym przyczynkiem takiego stanu rzeczy jest decyzja zaprzestania produkcji minikomputera MERA - 400. Decyzja ta spowodowała pewną niechęć kontynuowania rozwoju oprogramowania podstawowego dla tego minikomputera. Jednocześnie jednak, istnieje brak na krajowym rynku minikomputera o podobnych parametrach eksploatacyjnych i sprzętowych co MERA-400.

Dodatkowo; w chwili zaakceptowania i rozpowszechnienia systemu CROOK - 4 pojawiła się wizja możliwości powstania na bazie CROOK'a systemu UNIX, który obecnie jest powszechnym i obowiązkowym systemem na światowych rynkach komputerowych niezależnie od rodzaju oferowanego sprzętu.

Powodem powstania takiej wizji kontaktu z powszechnym już dzisiaj na świecie systemem operacyjnym UNIX jest ideowa zbieżność CROOK'a z UNIX'em. Podstawowymi różnicami pomiędzy tymi systemami to: odmienny rdzeń CROOK'a-4, CROOK-4 nie jest napisany w języku C oraz przede wszystkim brak kompilatora języka programowania C w systemie CROOK.

W opracowaniu /2/ można znaleźć ogólną prezentację i historię powstawania systemu UNIX.

Dzisiaj można stwierdzić, że jakiegokolwiek komputer, który nie posiada systemu operacyjnego opartego na UNIX'ie, nie posiada kompilatora języka C oraz oprogramowania podstawowego

napisanego w języku C nie może być przedmiotem oferty na światowych rynkach z prostej przyczyny: takiego komputera czy minikomputera nikt nie zakupi.

Rozpowszechnienie UNIX'a było możliwe dzięki takimi własnościami / w skrócie/:

- wysoka mobilność oprogramowania podstawowego systemu UNIX
- stosowany tylko język C,
- duża uniwersalność zastosowań systemu UNIX,
- duża prostota w obsłudze systemu,
- system UNIX otwarty na modyfikacje definiowane przez użytkownika,
- duża popularność języka programowania C.

2. Koncepcja polskiego UNIX'a.

Analizując wszystkie aspekty poruszanych spraw w pkt. 1 opracowania można stwierdzić, że istnieją dzisiaj wszystkie potrzebne atrybuty aby prototyp - model systemu UNIX powstały w kraju na bazie systemu CROOK-4 i minikomputerze MERA-400. Można nawet pokusić się na stwierdzenie, że jest to prawie idealny moment na rozpoczęcie prac nad opracowaniem oprogramowania podstawowego łącznie z systemem operacyjnym zgodnie z koncepcją UNIX'a.

Reasumując sprawy poruszone w pkt. 1 opracowania, można wymienić najpoważniejsze potrzebne atrybuty i zależności aby wizja powstania polskiego opracowania UNIX'a była jak najbardziej realna:

- a/ duży zasób wiedzy, duże doświadczenie prowadzonych prac nad ideowo zbieżnym systemie CROOK-4.
Badawczy styl i charakter prowadzonych prac zespołu informatyków z Instytutu Okrętowego Politechniki Gdańskiej.
- b/ możliwość i celowość poszerzenia głównego zespołu informatyków z Instytutu Informatyki UW, Politechnika Poznańska, Instytut Informatyki UJ, Instytut Ciężkiej Syntezy Organicznej i inne ośrodki.

- c/ doskonałe pole doświadczalne minikomputera MERA-400 - jest to jedyne środowisko sprzętowe w kraju, na którym powstawały i powstają doskonałe opracowania w zakresie oprogramowania podstawowego,
- d/ brak odpowiedniej maszyny na polskim rynku /prawdopodobnie co najwyżej 2-3 letni/ w znaczeniu powszechnym po-zaprzestaniu produkcji minikomputera MERA-400,
- e/ wysoka mobilność oprogramowania podstawowego według koncepcji UNIX spowoduje, że wyprodukowane przedmiotowe oprogramowanie będzie łatwo przenoszalne /w krótkim czasie i niewielkim nakładem sił/ na nowy komputer czy minikomputer.
- f/ istniejące prace nad kompilatorem języka C należy bezwzględnie przyśpieszyć poprzez powiększenie jednoosobowego obecnie "zespołu" do zespołu 2-3 osobowego. Po wykonaniu i homologacji kompilatora języka C przekazać go grupom projektantów opracowującym oprogramowanie podstawowe w języku C.
W ten sposób istnieje możliwość powstania w krótkim czasie podstawowego i koniecznego oprogramowania bibliotecznego kompilatora języka C - bez którego / z definicji C/ kompilator jest bezwartościowy i nie może istnieć,
- g/ świadomość tych wszystkich atrybutów /punkty a-f/ oraz ciekawość i brak UNIX'a w kraju powinna zmniejszyć niechęć projektantów do opracowywania nowych opracowań w zakresie oprogramowania podstawowego na minikomputerze MERA-400.

Realność przedsięwzięcia będzie wtedy realna, gdy opracuje się założenia i ogólny harmonogram przyszłych wspólnych prac. Te założenia i ogólny harmonogram prac muszą być wypracowane i zaakceptowane wspólnie w gronie osób zainteresowanych przedsięwzięciem.

Poza tym, bardzo ważnym czynnikiem musi być polityka finansowania tego przedsięwzięcia - zaakceptowana przez wszystkich i to wspólnie przez grono przyszłych projektantów.

3. Problemy użytkowników minikomputera MERA-400 eksploatujących system operacyjny CROOK-4.

Mimo dużego zainteresowania systemem CROOK-4 - duża ilość nabywców - w niewielu ośrodkach jak dotąd, zdecydowano się na wyłączną eksploatację tego systemu. Główną przeszkodą do wyłącznego korzystania z systemu CROOK-4 są trudności natury eksploatacyjnej, związanych z kłopotliwym przejściem z eksploatowanymi systemami użytkowymi, które projektowane były pod SOM-3.

W wielu ośrodkach pozyskano systemy wielodostępne SUM-3-podobne. Systemy te umożliwiają prawie natychmiastowe "przejście" eksploatowanych systemów użytkowych na nowy system. Jednak w chwili uporania się z tymi chwilowymi problemami i wstępnej korzyści z wielodostępu, zauważa się niedośkonalskość tych SOM-3 podobnych systemów, głównie z powodu przyjętych z założenia zasad architektury logicznej SOM-3.

W tym momencie najczęściej wzrasta się zainteresowanie systemem CROOK-4. Zazwyczaj w danym ośrodku kryształizuje się następujące rozwiązanie:

eksploatacja w SOM-3, natomiast opracowywanie nowych programów pod CROOK'iem - 4. Trudności wzmagają się w miarę wzrostu ilości opracowywanych nowych programów pod systemem CROOK-4 z dwóch podstawowych powodów:

- a/ z uwagi na odmienne organizacje zbiorów dyskowych -
- zwiększa się zapotrzebowanie na deficytowe urządzenia pamięci dyskowych i pakietów dyskowych,
- b/ w czasie eksploatacji SOM-3 brak dostępu do CROOK'a-4 i odwrotnie: w czasie pracy pod CROOK-4 nie można prowadzić normalnej pracy pod SOM-3.

Istniejący symulator SOM-3 pracujący pod systemem CROOK-4 jest rozwiązaniem /założenia/ niedoskonałym i jak dotąd nie spełnia oczekiwań użytkowników.

Praca pod tym symulatorem jest mało efektywna oraz wymaga wygospodarowania stosunkowo dużych obszarów pamięci dyskowej / nadal deficytowej/ Dla użytkowników posiadających jedną stację pamięci dyskowej /napewno zdecydowana większość/ praca pod symulatorem możliwa jest praktycznie tylko dla jednego systemu użytkowego w SOM-3.

Natomiast problem "przejścia" całkowicie z tzw. eksploatacją użytkową w SOM-3 na odmienny CROOK-4 urasta do dużych problemów organizacyjnych zwłaszcza wtedy, gdy weźmiemy pod uwagę ilość eksploatowanych programów użytkowych.

Programy te powinny bezwzględnie być nadal w eksploatacji i jednocześnie należy prowadzić prace związane z "przejściem" na inny system operacyjny - oczywiście wszystkie te prace muszą być wykonywane na tym samym sprzęcie MERA-400!

Taka "operacja" trwa często około roku czasu /z doświadczenia autora/ - praca w trakcie jej trwania mało efektywna powodująca jednocześnie "zamrożenie" wykonawstwa nowych opracowań w danym ośrodku. Często w trakcie dokonywania takiej "operacji" wąskim gardłem jest ograniczona przestrzeń adresowa pamięci dyskowej i należy często uciekać się do ekwilibrystycznych zabiegów scalania zbiorów dyskowych.

Podane ryżej uwarunkowania powodują duże zgłaszane zapotrzebowanie przez użytkowników na program/system umożliwiający automatyczne przekodowanie programów źródłowych napisanych w FORTRANIE IV S i ASSEMBLER SOM-3 lub FORTRAN z tzw. "wstawkami" ASSEMBLER, odpowiednio na FORTRAN I ASSEMBLER CROOK-4.

Z uwagi na wymagany krótki czas wykonawstwa takiego programu/systemu i jego z założenia krótki czas życia, program ten powinien być kodowany w języku BASIC / bardzo efektywny do takich celów/ i posiadający 2,3 przebiegi procesu tłumaczenia.

Czas procesu automatycznego tłumaczenia dokonywany przez przedmiotowy program / system jest mało istotny - istotny jest czas wykonania takiego opracowania i jego rozpowszechniania wśród użytkowników.

Koniecznym byłoby, aby po uzgodnieniu założeń projektowych i rozpoczęciu prac nad tym opracowaniem powiadomić wszystkich potencjalnych zainteresowanych - poza korzyściami oczywistymi, wyzwała się właściwe pozytywne bodźce dla wykonawców tego programu.

Dodatkowo, że program ten powinien być napisany w języku BASIC, powinien być dostarczany dla użytkowników tylko w wersji źródłowej. Istnieć będzie w ten sposób możliwość wprowadzenia poprawek przez lokalnych ekspertów.

4. Języki programowania - kompilatory.

W chwili obecnej i w najbliższej przyszłości dostępne będą tylko dwa języki programowania w systemie CROOK-4: BASIC, FORTRAN.

W stosunku do standardowego języka Basic w systemie CROOK-4 został on znacznie poszerzony o operacje na łańcuchach. Wobec wielkiej popularności i prostoty języka BASIC stosowanego powszechnie w mikrokomputerach osobistych znajduje on również szerokie zastosowanie w minikomputerze MERA-400. Jednakże BASIC nie jest ekonomiczny do budowania większych programów czy systemów zarówno do problemów inżynierskich, naukowo-technicznych czy przetwarzania danych, z uwagi na stosunkowo duży czas obliczeń i dużą pamięciochłonność.

Podstawową wadą systemu BASIC-CROOK jest brak jego dokumentacji eksploatacyjnej.

Natomiast FORTRAN JEST powszechnie znanym i stosowanym językiem, dla problemów inżynierskich i naukowo-technicznych Kompilator tego języka /wersja nr 30/

W systemie CROOK-4 jest już zainstalowany na wielku maszynach MERA-400 i można stwierdzić, że został on przyjęty przychylnie przez jego użytkowników.

Znakomitą zaletą tego kompilatora jest jego duża szybkość kompilacji oraz otrzymywany efektywny kompilot pod względem wielkości przestrzeni adresowej, przez co między innymi uzyskuje się krótkie czasy obliczeń zwłaszcza procedur arytmetycznych. W systemie minikomputera MERA-400 do tej pory nie był znany żaden z kompilatorów o tak wysokich parametrach eksploatacyjnych.

W opracowaniu /3/ znajdują się wyniki pomiarów czasów kompilacji i czasy działania programów testowych różnych kompilatorów /FORTAN-PASCAL, LOGLAN/ i różnych systemów operacyjnych minikomputera MERA-400.

Kompilator FORTAN-CROOK został oceniony najwyżej w tym opracowaniu. Nie budzi zastrzeżeń zakres i poprawność bibliotek oraz dokumentacja eksploatacyjna kompilatora.

Pewną wadą kompilatora FORTAN-CROOK są mało elastyczne z niezbyt dużymi możliwościami korzystania z procedur bezpośrednio dostępu do binarnych zbiorów dyskowych zarówno dla redagowanych i nieredagowanych rekordów. Wada ta wynika z przyjętej koncepcji obsługi zbiorów dyskowych w kompilatorze i z tego względu nie ma możliwości wyeliminowania tej wady bez gruntowej zmiany części kompilatora.

Następną wadą systemu FORTRAN-CROOK lecz nie wynikającą już z samego kompilatora jest translator języka symbolicznego ASSEMBLER, który jest stosowany w drugim przebiegu kompilacji języka FORTRAN na język wewnętrzny maszyny MERA-400. Konstrukcja tego translatora jest prosta i nie przewiduje ona stosowania konsolidacji relokowalnych modułów binarnych.

W systemie FORTRAN-CROOK konsolidację dokonuje się jeszcze przed etapem translacji na drodze łączenia modułów źródłowych w Języku symbolicznym ASSEMBLER.

Rozwiązanie to powoduje, że moduły do konsolidacji źródłowej muszą być przechowywane w postaci źródłowej języka symbolicznego, zajmując średnio kilkakrotnie większe rozmiary zbiorów dyskowych niż w przypadku stosowania relokowalnych zbiorów binarnych.

W konsekwencji czego, wszystkie zbiory bibliotek systemu FORTRAN-CROOK oraz własnych użytkowników, muszą być przechowywane w postaci źródłowej języka symbolicznego, zajmując średnio kilkakrotnie większe rozmiary zbiorów dyskowych niż w przypadku stosowania relokowanych zbiorów binarnych.

Następną wadę translatora języka symbolicznego ASSEMBLER podobnie jak BASIC jest brak dokumentacji eksploatacyjnej. Chcąc tę wadę wyeliminować należałoby opracować nowy translator języka symbolicznego łącznie z konsolidatorem. Przy czym nowy translator języka symbolicznego powinien być oparty na identycznej sybolice kodowania instrukcji jak w istniejącym ASSM.

Nie spełnienie tego wymogu spowodowałoby brak możliwości współpracy obecnego kompilatora FORTRAN'u z nowym translatorem.

Z przedstawionego przeglądu języków programowania wysokiego poziomu systemu CROOK-4 wynika przede wszystkim, że istniejącą bazą językową jest bardzo szczupła oraz istnieje pewna dotkliwa wada braku translatora języka symbolicznego łącznie z konsolidatorem.

Porównując istniejącą bazę językową systemu CROOK-4 z systemami SOM-3 podobnymi wynika, że rozwój kompilatorów języków wysokiego poziomu dla CROOK'a jest konieczny, chociażby powodowany koniecznością wymiany oprogramowania pomiędzy użytkownikami obu tych systemów: SOM-3 i CROOK-4. Inicjatywa wypracowana na naradzie roboczej projektantów oprogramowania podstawowego, która odbyła się w październiku 1984 w Gdańsku /4/ powinna być kontynuowana w zakresie wymiany kompilatorów języków PASCAL, LOGLAN istniejących pod SOM-3 oraz kompilatora języka C pod CROOK'iem.

Prowadząc dyskusję nad rozwojem bazy językowej systemów CROOK-4 i SOM-3 dokonano następujących założeń:

- wybór języka programowania powszechnie znanego i stosowanego w kraju i na świecie;

- poszerzyć możliwości zastosowań dziedzinowych minikomputera MERA-400,
- uwzględnić sprzętowe możliwości typowych zestawów MERA-400: pamięć operacyjna, przestrzeń adresowa pamięci dyskowych oraz możliwości rozbudowy minikomputera MERA-400,
- zapotrzebowania użytkowników,
- czas niezbędny na opracowanie i rozpowszechnienie kompilatora.

Analizując dziedziny zastosowań systemów informatycznych użytkowników MERY-400, wynika, że znakomita większość zastosowań istniejących to oprogramowanie inżynierskie, naukowo-techniczne i automatyzacja prac projektowych.

W tych zastosowaniach bardzo przydatny jest język programowania FORTRAN.

Mimo, że FORTRAN jest najstarszym językiem programowania wysokiego poziomu, w naszym kraju i na świecie jest powszechnie znanym i stosowanym oraz większość istniejących opracowań w tych dziedzinach niezależnie od sprzętu jest oprogramowana w tym języku.

Znanych jest wiele języków programowania /5/ wysokiego poziomu zorientowanych problemowo w omawianych dziedzinach i bardziej uniwersalnych: LOGLAN, MODULA, lecz są to języki programowania rozbudowane przez co są bardzo uniwersalne do wielu zastosowań.

Natomiast istnieje coraz większe zapotrzebowanie użytkowników MERY-400 na język programowania wysokiego poziomu do celów przetwarzania danych - systemów wspomagania zarządzaniem: systemy bankowe, magazynowe, handlowe, administracyjne itp.

Językiem programowania zorientowanym problemowo do tych celów jest COBOL.

Jest to język prosty, powszechnie znany i stosowany w kraju i na świecie - został opublikowany w 1959 r.

COBOL jest językiem programowania, który został specjalnie opracowany właśnie do problematyki przetwarzania da-

nych do budowy systemów informatycznych wspomaganie zarządzaniem. Język COBOL nie posiada do tej pory konkurencyjnego języka programowania do przetwarzania danych.

Natomiast wadą tego języka jest jego mała uniwersalność zastosowań - lecz wada ta staje się szczególnie zaletą w aspekcie przedstawionych powyżej problemów.

Do najważniejszych cech języka COBOL należą:

- łatwość specyfikowania zbiorów danych;
- proste metody operacji na zbiorach danych i ich elementach;
- prosta arytmetyka kodowa znakowo-tylko 4 działania arytmetyczne,
- stosowana arytmetyka kodowana znakowo pozwala na operowaniu dużymi i małymi liczbami bez zaokrągleń czyli zmiany dokładności;
- łatwość w programowaniu;
- mała uniwersalność w zastosowaniach.

Niebagatelną zaletą języka COBOL w omawianym aspekcie rozwoju oprogramowania podstawowego MERY-400 jest przewidywany krótszy czas opracowania kompilatora tego języka.

5. Biblioteka testów technicznych systemu CROOK-4

System CROOK - 4 spowodował, że minikomputer MERA-400 czyli procesor, pamięć operacyjna pamięci dyskowe, inne pamięci zewnętrzne oraz urządzenia znakowe mogą być w pełni wykorzystane. Poza tym, wobec uniwersalności systemu, użytkownicy rozbudowują środowisko sprzętowe MERY-400, głównie poprzez pamięci operacyjne, pamięci dyskowe i terminale.

Wobec możliwości pracy pod CROOK'iem /wielodostęp, wieloprogramowość/ oraz rozbudowanego często sprzętu minikomputer MERA-400 stał się jakościowo zupełnie nową maszyną niż w przypadku stosowanego SCM-3

Stan powyższy, możliwego większego wykorzystania sprzętu powoduje, że awaryjność zestawu MERA-400 staje się bardzo istotnym problemem technicznym, organizacyjnym i ekonomicznym. Każdy przestój maszyny spowodowany awaryjną czy niepewną pracą jest teraz bardziej dotkliwy i z ekonomicznego punktu widzenia bardziej kosztowny. Chodząc z większej ilości użytkowników czy starego problemu w naszym kraju - części zamiennel.

Wobec powyższych względów, możliwość sprawdzenia stanu gotowości techniczno-eksploatacyjnej minikomputera MERA-400 bez wyłączenia go z eksploatacji czyli zatrzymania pracy systemu operacyjnego nader istotny.

Możliwość diagnozowania maszyny bez wyłączania jej z eksploatacji - czyli w czasie gdy wszyscy lub część użytkowników pracuje i eksploatuje swoje programy jest bardzo ważny.

Możliwość takiego diagnozowania zestawu maszyny MERA-400 - nie był dotąd znany. Diagnozowanie czy testowanie zestawu komputerowego pod działającym systemem operacyjnym czyli w czasie normalnej eksploatacji jest szeroko znany i stosowany : maszyny cyfrowe wzorowane na architekturze logicznej i oprogramowaniu maszyn ICL-1900, np. seria komputerów ODRA 1300 /1304 ;1305; 1325/.

Praktycznie wszystkie maszyny średnich i dużych mocy obliczeniowych posiadają taką możliwość diagnozowania - posiadają one tzw. bibliotekę oprogramowania technicznego działającą pod systemem operacyjnym i bliźniaczą lecz często rozbudowaną na maszynę bez systemu operacyjnego.

W przypadku MERY - 400 istnieje znany pakiet oprogramowania technicznego działającego tylko i wyłącznie bez systemu operacyjnego - STM /System Testów MERRY-400/

System tych testów był opracowany u producenta MERY-400 w czasie gdy jedynym znanym i stosowanym systemem operacyjnym był SOM - 3.

W czasie powszechnej eksploatacji SOM-3 maszyny MERA-400 nie były w tak wysokim stopniu wykorzystywane jak obecnie - pod systemami operacyjnymi wielodostępnymi z możliwością

pracy wieloprogramowej.

Dodatkowo maszyny działające pod SOM-3 nie były rozbudowane sprzętowo czyli potrzeba stosowania wyspecjalizowanego oprogramowania technicznego pod systemem operacyjnym była niewielka.

Ponadto, występowała niemożność stosowania takiego sposobu diagnozowania MERY-400 pod systemem SOM-3 z prostej przyczyny - istniała możliwość pracy tylko jednego użytkownika i jego jednego programu!

Czyli w czasie pracy programu użytkowego nie było możliwości pracy innego drugiego programu użytkowego czy testującego maszynę.

Dopiero po powszechnym stosowaniu systemów wielodostępnych taka możliwość istnieje czy wręcz jest konieczna z powodu rozbudowanego sprzętu.

Jeszcze jednym powodem do zwrócenia uwagi na testowanie sprzętu w czasie pracy systemu operacyjnego jest znana niedoskonała oprogramowana biblioteka testów uruchomieniowych STM - jak dotąd jedyna!

Według proponowanej koncepcji, należy spowodować opracowanie takiej biblioteki testów technicznych pod systemem CROOK-4. Jest ona w chwili obecnej niezbędnie potrzebna co zostało stwierdzone w wielu instalacjach MERY-400 działających pod CROOK'iem.

Testy z biblioteki STM pracowały poprawnie!

Proponuje się opracowanie programów - testów do diagnozowania: procesora, tzw. przystawki ZMP, pamięci operacyjnej, pamięci dyskowych oraz urządzeń znakowych - terminali /drukarki z klawiaturą i monitory ekranowe/ oraz programy testujące współdziałanie wszystkich urządzeń w kanale pamięciowym.

Pewnym wzorem do opracowania takiej biblioteki testów technicznych może być biblioteka testów stosowana w maszynach ODRA-1300 - zwłaszcza ODRA-1305.

Zaletą takiego oprogramowania specjalistycznego jest możliwość korzystania z takich testów przez każdego użytkownika systemu CROOK-4 bez wyłączania eksploatacji - czyli w czasie normalnej pracy.

6. Wnioski

W przedstawionej koncepcji rozwoju oprogramowania podstawowego minikomputera MERA-400 ujętej w tym opracowaniu można wydzielić dwie tendencje rozwojowe:

a/ bezpośrednia /krótkofalowa/ - w opracowaniu podano minimum prac, które należałoby podjąć aby system CROOK-4 pozbył się obecnych wad - zbyt szczupłego w stosunku do SOM-3 oprogramowania podstawowego.

Reasumując, autor opracowania postuluje podjęcia następujących głównych prac: kompilatory języków programowania istniejących w SOM-3: PASCAL i LOGLAN, kompilatora języka COBOL, nowego translatora języka symbolicznego ASSEMBLER łącznie z konsolidatorem oraz programu, który pozwoliłby na automatyczne przekodowywanie źródłowych programów FORTRAN/SOM-3 łącznie z tzw. wstawkami języka symbolicznego odpowiednio na FORTRAN/CROOK-4. Do tej grupy rozwoju oprogramowania podstawowego zalicza się

b bibliotekę oprogramowania technicznego działającą pod systemem CROOK-4.

b/ perspektywiczna /długofalowa/ - podjęcie prac nad prototypem systemu UNIX na bazie rodziny systemów CROOK i bazie minikomputera MERA-400. W tej części najważniejszą sprawą jest uzyskanie możliwie jak najszybciej sprawnego kompilatora języka C łącznie z biblioteką procedur bez których kompilator jest mało wartościowy. Dopiero mając do dyspozycji sprawny kompilator języka C można przystąpić do właściwej realizacji prototypu systemu i koncepcji systemu UNIX

Tylko na pozór obydwie tendencje zostały tutaj wyspecyfikowane oddzielnie. W ten sposób można wykazać, że koncepcja bezpośrednia i perspektywiczna nie mogą istnieć oddzielnie, nie mogą być realizowane oddzielnie czy wręcz koncepcją bezpośrednią implikuje perspektywiczna i na odwrót.

Realizowanie tylko części bezpośredniej czyli pozbywanie systemu CROOK-4 nad wobec decyzji zaprzestania produkcji minikomputera MERA-400 byłoby dużym błędem. W ten sposób cały wysiłek włożony do zrealizowania systemu CROOK i jego oprogramowania poszedłby na marne. Czas życia tego całego oprogramowania łącznie z systemem CROOK-4 byłby krótki prawdopodobnie kilkuletni - upadek tego dorobku nastąpiłby w chwili pojawienia się odpowiedniego minikomputera jako powszechnego sprzętu na polskim rynku. Byłaby to wielka szkoda w rozwoju informatyki w naszym kraju wobec faktur istnienia systemu CROOK-4 ideowo zbieżnego do obecnie już powszechnego systemu UNIX i jego koncepcji.

Najważniejszą zaletą koncepcji systemu UNIX jest jego łatwa przenoszalność na dowolny sprzęt komputerowy. Ta zaleta, głównie spowodowała bardzo duże spopularyzowanie systemu UNIX w świecie w ostatnich latach. Dzisiaj praktycznie nie można na rynkach międzynarodowych dokonać transakcji sprzedaży komputerów, nie tylko średnich mocy obliczeniowych, który nie posiadałby systemu UNIX.

Te wszystkie zależności powodują, że przedstawiona koncepcja rozwoju oprogramowania podstawowego minikomputera MERA-400 jest spójna i możliwa do realizacji.

Co więcej, istnieje możliwość realizacji tego programu działania jakby dwutorowa.

Po zaktywizowaniu prac nad kompilatorem języka C /pkt.1 opracowania/ Możliwe jest szybkie jego rozpowszechnienie przede wszystkim projektantom oprogramowania podstawowego MERA-400. W ten sposób możliwa jest czy nawet konieczna realizacja równoległa przedstawionej w opracowaniu koncepcji:

programu bezpośredniego i całego długofalowego. Tej szansy zmarnować nie można

7. Literatura

- 1/ Alan C. Schaw - Projektowanie logiczne systemów operacyjnych - WNT - Warszawa 1980 r.
- 2/ T. Rawiński - "System operacyjny UNIX - jednolita w skali międzynarodowej podstawa zastosowań komputerów". Praca wykonana w ramach tematu BW-845128 - Politechnika Gdańska, Instytut Okrętowy.
- 3/ B. Machowiak - Analiza porównawcza systemów operacyjnych dla minikomputera MERA-400" Praca wykonana w ramach prac Klubu Użytkowników Minikomputera MERA-400. BISTYP- Warszawa 1984
- 4/ B. Machowiak - Wyniki narady roboczej projektantów systemów operacyjnych MERY-400
Referat wygłoszony na II Konferencji Użytkowników Minikomputera MERA-400 - Gdańsk - październik 1984 r
- 5/ Zeszyty INFORMATYKA 1983/1984 - zbiór artykułów /6egz/
Przegląd języków wysokiego poziomu. Prognozy rozwoju.

Andrzej Ziembiewicz

Analiza Charakterystycznych Cech Systemów Operacyjnych CROOK-4 i UNIX

Spis treści

1. Wprowadzenie
2. System zbiorów
 - 2.1. Struktura systemu zbiorów
 - 2.2. Implementacja
3. Zarządzanie procesami
 - 3.1. Pojedynczy proces
 - 3.2. Procesy spokrewnione i współpracujące
4. Podsumowanie

1. Wprowadzenie

UNIX jest obecnie jednym z najbardziej rozpowszechnionych systemów operacyjnych na świecie, spośród krajowych systemów zaś CROOK-4 jest najbardziej do niego zbliżony pod względem ogólnej koncepcji.

Niestety są one w kraju raczej mało znane z powodu braku pozycji na ich temat w polsko-języcznej literaturze fachowej. Autor niniejszego pozwolił sobie zatem znacznie przekroczyć przewidziane dla referatu ramy (za co z góry przeprasza) w przekonaniu, że jest to dobra okazja do przybliżenia tych dwóch nowoczesnych systemów operacyjnych szerszemu ogółowi.

Nie jest to pełna analiza wszystkich istotnych własności tych systemów, a tylko najbardziej podstawowych, lub szczególnie interesujących. Najbardziej podstawową wspólną ich cechą jest to, (specjaliści zechcą wybaczyć znacznie uproszczenie) że w zasadzie każdy obiekt statyczny w systemie jest zbiorem, zaś każdy obiekt dynamiczny jest procesem.

Każdy proces (upraszczając) można uważać za instancję pewnej funkcji, odwzorowującej zbiór w zbiór, zaś cały system operacyjny - za kolekcję takich funkcji. Totem dla pobieżnej analizy wydaje się być wystarczające omówienie systemu zbiorów oraz zarządzania procesami w obydwu systemach.

2. System zbiorów

System zbiorów jest wbudowany w jądro systemu operacyjnego (zarówno CROOK-4 jak i UNIX). W przypadku systemu zbiorów termin 'system' rozumiemy jako parę (zbiór elementów, struktura tego zbioru). Elementami są zwykle zbiory dyskowe, urządzenia zewnętrzne, linie komunikacyjne itp. obiekty. W ogólności każdy obiekt fizyczny w danej instalacji komputerowej (włącznie z pamięcią operacyjną) może być widziany jako zbiór.

Zbiór jest identyfikowany poprzez nazwę. Użytkownika w zasadzie nie interesuje fizyczna lokalizacja zbioru na nośniku. Co do zawartości i struktury wewnętrznej zbioru system nie czyni żadnych założeń. Mówiąc dokładniej, system traktuje zawartość zbioru wyłącznie jako ciąg bajtów, o takiej długości, jaka została doń zapisana. Długość zbioru rośnie automatycznie w miarę dopisywania informacji. Wszystkie własności omówione wyżej dotyczą obydwu systemów operacyjnych.

2.1. Struktura systemu zbiorów

System zbiorów posiada strukturę drzewiastą. Węzłami są skorowidze, zaś liśćmi - zbiory. Skorowidz zawiera nazwy węzłów (liści) bezpośrednio mu podległych. Głębokość takiej hierarchii nie jest ograniczona. W czasie pracy użytkownik wybiera dowolny z dostępnych dla niego skorowidzów jako bieżący. Wybór ten obowiązuje do chwili zmiany (na życzenie użytkownika). Poniżej omówiono własności, które różnią się w obu systemach.

Struktura globalna.

UNIX: Strukturę stanowi jedno drzewo, którego korzeń znajduje się na dysku systemowym. Jeżeli w instalacji znajduje się więcej nośników, to na każdym z nich znajduje się drzewo systemu zbiorów, jednak przez system jest ono widziane dopiero wtedy, gdy zostanie włączone jako poddrzewo do drzewa głównego.

CROOK: Strukturę stanowi uporządkowana rodzina drzew. Każdy nośnik (np. talerz dyskowy, dyskietka) posiada własne drzewo. Poza numerem porządkowym nośnik posiada nazwę.

Użytkownicy.

UNIX: System prowadzi jeden skorowidz użytkowników, znajdujący się na systemowym nośniku. Każdy użytkownik może założyć obcy nośnik i do jednego ze swoich skorowidzów dołączyć znajdujące się na nim podrzewo, stając się w ten sposób jego właścicielem. W ten sposób ochrona zawartości nośnika polega na trzymaniu go w zamkniętej szafie.

W systemie UNIX skorowidz użytkowników jest jednopoziomowy, tzn. nie jest możliwe utworzenie hierarchii użytkowników. Przy pracy zespołowej jest to pewna niewygodą, którą system próbuje złagodzić wprowadzając pojęcie 'grupy użytkowników', w ramach której można zadeklarować rozszerzenie kryteriów dostępu do pewnych zbiorów, używanych przez członków grupy. Nie rozwiązuje to jednak sprawy w pełni.

CROOK: Skorowidz użytkowników prowadzony jest przez system na każdym nośniku niezależnie. Ochrona jest w ten sposób bardziej pewna, bowiem zakładając obcy nośnik użytkownik ma dostęp tylko do tych zbiorów znajdujących się na nim, które są udostępnione przez ich właściciela innym użytkownikom.

W systemie tym możliwe jest tworzenie hierarchii użytkowników, tj. użytkownik może utworzyć podległych sobie pod-użytkowników. Ponadto tworzonemu pod-użytkownikowi można przydzielić limit łączny przestrzeni na nośniku (udzielając go z własnego limitu), co jest własnością istotną w warunkach dostępnych u nas nośników o bardzo małej pojemności.

Widzialność i identyfikacja zbioru.

UNIX: Zbiór identyfikowany jest za pomocą nazwy. Szukanie zbioru przez system odbywa się tylko w jednym skorowidzu i rozpoczyna się od wyznaczenia tego skorowidza.

Jeśli nazwa nie zawiera znaków '/', to jest ona nazwą końcową, a zbiór szukany jest tylko w skorowidzu bieżącym. W przeciwnym razie jest to pełna nazwa zbioru (tzw. nazwa ścieżkowa), w której część tekstu za ostatnim znakiem '/' jest nazwą końcową zbioru, zaś część poprzedzająca jest ciągiem nazw skorowidzy, które przeszukiwane są kolejno. Ostatni z nich jest skorowidzem docelowym, w którym szukany jest zbiór.

Wyznaczenie ścieżki rozpoczyna się dwójako; od korzenia systemu zbiorów gdy pierwszym znakiem nazwy ścieżkowej jest '/', lub od skorowidza bieżącego w przeciwnym razie. Np. nazwa 'dir1/dir2/file' powoduje, że w skorowidzu bieżącym szukana jest pozycja odpowiadająca skorowidzowi 'dir1', w tym skorowidzu następnie pozycja odpowiadająca skorowidzowi 'dir2'. Jest to właściwy skorowidz, w którym teraz szukany jest zbiór 'file'.

Algorytm ten w pierwszej chwili sprawia wrażenie zawilego, jest jednak bardzo prosty i posiada szereg zalet. Wszystkie nazwy zbiorów i skorowidzów są lokalne i bez wskazania ścieżki są widzialne tylko z bezpośrednio nadrzędnego skorowidza. Te same nazwy mogą się powtarzać na różnych piętrach systemu zbiorów i u różnych użytkowników nie powodując konfliktów ani niejednoznaczności.

Z drugiej strony znalezienie zbioru znajdującego się w dolnych piętrach drzewa wymaga przeszukania dużej liczby skorowidzów i jest czasochłonne.

CROOK: Zbiór identyfikowany jest nazwą, która może być co najwyżej trój-składnikowa. Nazwa nie zawierająca znaku '/' jest nazwą końcową zbioru. W przeciwnym razie jest to pełna nazwa zbioru (tzw. tytuł zbioru), gdzie pierwszym składnikiem jest nazwa nośnika, zaś drugim - nazwa skorowidza. Np. nazwa 'SYS/BIBL/PROGI' oznacza zbiór 'PROGI', znajdujący się na nośniku 'SYS' pod skorowidzem 'BIBL'.

Poszczególne składniki tytułu zbioru mogą być puste. Pusta nazwa skorowidza oznacza skorowidz bieżący, zaś pusta nazwa nośnika powoduje, że przeszukiwane są kolejne nośniki aż do znalezienia najbliższego, na którym znaleziony zostanie wskazany skorowidz i zbiór.

W rozwiązaniu tym interesująco przedstawia się sprawa widzialności zbioru w przypadku, gdy nie został wskazany skorowidz. Obowiązuje tu zasada taka sama, jak w przypadku widzialności identyfikatorów w językach programowania o strukturze blokowej. Tj. zbiór będący pozycją w danym skorowidzu jest widzialny w całym poddrzewie tego skorowidza, z wyłączeniem tych skorowidzów, w których jest on zasłonięty przez zbiór lokalny o tej samej nazwie. (Dotarcie do zasłoniętego zbioru jest możliwe przez podanie tytułu zbioru).

Algorytm ten jest bardzo elegancki, powoduje jednak pewne skutki uboczne. Jednym z nich jest wymaganie, aby skorowidze posiadały jednoznaczne nazwy w ramach całego systemu zbiorów. Prawdopodobnie warunek ten dałby się osłabić. To jednak pociągnęłoby ograniczenia w sięganiu do zbiorów znajdujących się w innym poddrzewie, bądź ograniczenia w zmianach skorowidza bieżącego.

Warto też zauważyć, że choć pewną niewygodą jest niemożliwość utworzenia skorowidza o żądanej nazwie, gdy inny użytkownik już taki posiada, jest to kompensowane tym, że dzięki jednoznaczności nazw skorowidzów wyszukiwanie zbiorów jest bardzo szybkie.

Analizie identyfikacji i widzialności zbiorów poświęciliśmy stosunkowo dużo miejsca, ponieważ z punktu widzenia użytkownika pod tym względem oba systemy zbiorów różnią się w sposób najbardziej widoczny. Inne własności różnią je bardziej 'zakulisowo' i w sposób mniej zauważalny.

Zbiory robocze (chwilowe).

UNIX: Zbiory robocze nie są w żaden sposób wyróżnione. Aby uniknąć zaśmiecania nośnika, przewidziany jest specjalny skorowidz '/tmp' ogólnie dostępny dla zapisu, w którym użytkownicy i programy mogą tworzyć zbiory chwilowe. Skorowidz ten jest czyszczony periodycznie przez kierownika instalacji (ręcznie), bądź przez procedurę startu systemu operacyjnego (automatycznie).

CROOK: Wyróżniona jest kategoria zbiorów chwilowych. Zbiór utworzony jako chwilowy zostaje automatycznie skasowany przez system operacyjny z chwilą zakończenia sesji użytkownika. Zapobiega to zaśmiecaniu nośnika, jeśli zaś chcemy zbiór chwilowy zachować, należy zażądać jego zamiany na zbiór trwały.

2.1. Implementacja

Wewnętrzna realizacja obu systemów zbiorów różni się w znacznie większym stopniu, niż ich ogólna koncepcja i własności użytkowe. Ponieważ różnice te dotyczą poziomów w zasadzie niewidocznych dla przeciętnego użytkownika systemu, niniejszy paragraf jest adresowany do węższej grupy czytelników, interesujących się budową systemów operacyjnych.

Nośniki używane w obu systemach są formatowane w sektory stałej długości (zwykle 512 bajtów). Zbiór sektorów jest zorganizowany w ciąg bloków logicznych numerowanych kolejno poczynając od 0. Blok jest najmniejszą jednostką informacji przesyłaną w jednej transmisji i zawiera jeden lub kilka kolejnych sektorów (zależnie od implementacji). System zbiorów i wszystkie wyższe warstwy oprogramowania korzystają z bloków logicznych.

Implementacja pojedynczego zbioru

Najistotniejszą własnością różniącą oba systemy jest ciągłość zbioru na nośniku.

UNIX: Obszar zajmowany przez zbiór na nośniku jest nieciągły. Jeżeli w czasie pisania do zbioru ulega on powiększeniu, wyszukany zostaje najbliższy wolny blok na nośniku i dołączony do zbioru. Przy skasowaniu zbioru wszystkie należące doń bloki zostają zwrócone do puli wolnych bloków:

Główną zaletą tego rozwiązania jest to, że nie zachodzi konieczność odzyskiwania nieużytków, kompresji nośnika itp. operacji, kłopotliwych w innych systemach. Każdy zbiór daje się zawsze powiększyć, jeśli na nośniku znajduje się choć jeden wolny blok.

Słabą stroną zaś jest niekorzystny wpływ na szybkość działania całego systemu. Przeczytanie zbioru, a także wywołanie programu, wymagające przeczytania binarnego zbioru zawierającego obraz programu, wymaga wykonania tylu transmisji, w ilu rozłącznych obszarach znajdują się bloki zbioru. Ostatnio rozwój systemu UNIX ma tendencję do wydłużania wielkości bloków lub czynienia innych kroków w kierunku większej ciągłości zbiorów.

CROOK: zbiór zajmuje zawsze ciągły obszar na nośniku. Przy dopisywaniu informacji do zbioru zostaje zarezerwowany pewien dodatkowy obszar znajdujący się za nim. W przypadku braku takiego obszaru cały zbiór zostaje przeniesiony w inne miejsce.

Zaletą tego rozwiązania jest oczywista. Cały zbiór zawsze daje się przeczytać za pomocą jednej transmisji. Zapewnia to dużą szybkość działania systemu. Z drugiej strony powierzchnia nośnika ulega stopniowo fragmentacji. Co pewien czas trzeba przeprowadzać operację kompresji w celu odzyskania nieużytków i złączenia ich w spójny obszar. Częstość z jaką musi być przeprowadzana ta operacja zależy od specyficznych właściwości zastosowań danej instalacji. Na ogół nie jest to konieczność zbyt dokuczliwa, ponieważ algorytm alokacji nowych zbiorów ma pewną naturalną tendencję do zapełniania powstających 'dziur'.

Implementacja całości struktury.

W obu systemach głównym elementem jest tablica (lista) metryk zbiorów. Pozycję w tablicy stanowi metryka pojedynczego zbioru, zawierająca informacje o jego atrybutach i lokalizacji na nośniku. Szczegóły rozwiązań są w obu systemach całkowicie odmienne.

UNIX: Sama tablica (nazywana tu 'i-list') nie jest zbiorem. Znajduje się w zastrzeżonej (służbowej) części nośnika i nie jest objęta drzewem struktury. Pozycja zawiera identyfikator użytkownika - założyciela zbioru, datę założenia, ostatniego dostępu i ostatniej modyfikacji, kryteria dostępu i in. W szczególności zawiera też dziesięć numerów pierwszych dziesięciu bloków zbioru, odsyłacz do tablicy pośredniej, zawierającej 128 następnym numerom bloków, odsyłacz do tablicy zawierającej odsyłacze do następnych 128 tablic i odsyłacz do podobnej tablicy, lecz z trzykrotnym pośrednim adresowaniem. Pozycja nie zawiera natomiast nazwy zbioru.

Drugim składnikiem systemu zbiorów jest drzewo struktury. Węzłami tego drzewa są skorowidze, które w tym systemie są zbiorami wyróżnionego typu, natomiast liśćmi są pozostałe zbiory. Skorowidze służą wyłącznie do opisanie struktury systemu zbiorów i nie zawierają żadnych informacji o samych zbiorach. Pozycję w skorowidzu stanowi para, której pierwszym elementem jest nazwa zbioru, drugim zaś - indeks metryki zbioru w tablicy metryk.

Z powyższego opisu łatwo zorientować się, że droga do informacji zawartej w zbiorze wskazanym przez użytkownika jest długa i prowadzi poprzez wiele transmisji (dyskowych). Przede wszystkim należy dotrzeć do skorowidza, w którym znajduje się wskazany zbiór. Po odszukaniu w nim nazwy zbioru należy odczytać indeks do tablicy metryk i pobrać metrykę zbioru. W razie potrzeby należy teraz odczytać jedną, dwie lub trzy tablice pośrednie z numerami bloków (zależnie od ilości pośrednich adresowań). Na koniec można odczytać blok zawierający potrzebny fragment zbioru. Dla dotarcia do następnego bloku w zbiorze całą operację należy powtórzyć. Toteż w systemie tym stosowany jest rozbudowany i dosyć zawiły mechanizm buforowania. Przechowywane są w pamięci operacyjnej wszystkie informacje o aktualnie używanych zbiorach, fragmenty tablicy metryk itp. W konsekwencji system jest pamięcio-żłonny i do sprawnego działania wymaga co najmniej 64KB. Mimo tych usprawnień czas składania zawartości zbioru w całość jest wyraźnie zauważalny i reakcja systemu na zlecenie uruchomienia programu wynosi czasami kilka sekund.

Zaletą zaś tego sposobu implementacji jest duża elastyczność w manipulowaniu zbiorami i skorowidzami. Co więcej, możliwy jest tu mechanizm tzw. 'wielokrotnych linków'. Polega on na tym, że w różnych skorowidzach (nawet należących do różnych użytkowników) można otwierać zbiory, związane z tym samym fizycznym miejscem na nośniku. Wystarczy w tym celu, aby dla zbiorów tych odpowiadały im pozycje w skorowidzach zawierały ten sam indeks metryki.

CROOK: Całość informacji o zbiorach znajduje się w tablicy metryk, która w systemie tym jest również zbiorem o ustalonej nazwie. Jest ona zorganizowana jako tablica z kluczem mieszającym, zaś całość rozwiązania podporządkowano wymaganiu, aby wyszukiwanie zbioru było jak najszybsze. Drzewo skorowidzów jest tu także zbiorem.

Gospodarka nośnikiem.

Gospodarka nośnikiem jest w obu systemach bardzo podobna. Nośnik jest podzielony na kilka części, traktowane przez system w odmienny sposób.

UNIX: Nośnik jest na ogół podzielony na 4 części (zwane tu partycjami). Pierwszą część stanowi blok o numerze 0, przeznaczony dla bootstrapu. Drugą jest blok o numerze 1, tzw. 'super-blok', zawierający opis podziału nośnika na partycje. Trzecią część stanowi obszar 'służbowy', zawierający m.in. tablicę metryk zbiorów. W przypadku, gdy nośnikiem tym jest systemowy talerz dyskowy, w obszarze tym znajduje się też przestrzeń zarezerwowana na obrazy programów, podlegających wymianom.

Czwartą i ostatnią partycją jest obszar przeznaczony na zbiory, w tym także skorowidze.

CROOK: Nośnik składa się najczęściej z trzech części. Pierwsza i druga są to obszary 'służbowe' i zawierają kilka wyróżnionych zbiorów, nie podlegających przenoszeniu w inne miejsce. Pierwszy z tych zbiorów o nazwie 'LABEL' jest metryką całego nośnika i opisuje jego podział. Drugim jest zbiór 'SYSTEM', zawierający system operacyjny. Druga część zawiera zbiór stanowiący skorowidz skorowidzów - 'DICDIC', oraz zbiór będący tablicą metryk zbiorów - 'FILDIC'. Trzecia część jest przeznaczona na pozostałe zbiory.

W ten sposób cała powierzchnia nośnika jest objęta systemem zbiorów. Jak już wspomnieliśmy, alokacja miejsca dla zbioru na nośniku jest poza gestią użytkownika. Jednakże w przypadku zaistnienia takiej potrzeby, użytkownik posiadający odpowiednie uprawnienia może uzyskać dostęp do wskazanego fizycznego miejsca na nośniku. W tym celu w systemie zbiorów na każdym nośniku generowana jest metryka pseudo-zbioru o nazwie 'GLOBAL', obejmującego całą przestrzeń nośnika, tak jakby to był jeden zbiór.

Autor niniejszego miał przyjemność przez parę ostatnich lat używać obu systemów. Doświadczenie to pozwala stwierdzić, że jeśli chodzi o systemy zbiorów, obydwa są bogate, wygodne w użyciu i własności żadnego z nich nie dają mu przewagi nad drugim. W systemie CROOK użytkownik jest jakby nieco bliżej sprzętu, który jest w większym stopniu 'widzialny' poprzez system. Sprawia też wrażenie bardziej zwanterego i przeciętnemu użytkownikowi łatwiej dotrzeć do różnych informacji systemowych, np. dokonać zmian konfiguracji sprzętu.

UNIX natomiast sprawia wrażenie większego i bardziej rozczłonkowanego. Informacje systemowe są rozrzucone po różnych zbiorach, znajdujących się w różnych skorowidzach i użytkownikowi łatwiej się w nim zgubić.

3. Zarządzanie procesami

W obu systemach proces jest rozumiany tak jak proces sekwencyjny w sensie matematycznym. W uproszczeniu można wyobrazić sobie, że zarządzanie procesami realizowane jest w ten sposób, iż dla wykonania każdej czynności (np. zlecenia użytkownika) powoływany jest nowy proces, który następnie zostaje skasowany po wykonaniu zleczonej czynności.

Każdy proces może powoływać procesy potomne. W ten sposób zlecona czynność może być wykonana nie przez pojedynczy proces, ale przez całą rodzinę spokrewnionych procesów. Dyskusję rozwiązań rozpoczniemy jednak od prostszego przypadku, gdy czynność jest wykonywana przez pojedynczy proces. Dla ustalenia uwagi przyjmijmy, że jest to wywołanie programu użytkownika.

3.1. Pojedynczy proces

W każdym wieloprogramowym systemie operacyjnym (nie tylko w systemach tu dyskutowanych) powołanie procesu wymaga zapewnienia przez system czterech elementów. Są to: pewien obszar pamięci operacyjnej zaalokowany dla procesu, obszar w pamięci masowej przeznaczony na przechowywanie obrazu pamięci operacyjnej procesu w czasie gdy podlega on wymianie, 'tablica stanu procesu' służąca do przechowywania przez system operacyjny informacji o procesie, niezbędnych zarówno w czasie, gdy proces jest wykonywany, jak również w czasie gdy proces został zawieszony i podlega wymianie, na koniec zaś tzw. środowisko procesu, a ściślej mówiąc zorganizowanie mechanizmów komunikacji procesu z jego otoczeniem.

Obszar pamięci operacyjnej procesu.

UNIX: Proces otrzymuje pewien obszar pamięci złożony z trzech części (segmentów). Pierwszym z nich jest tzw. segment kodu, w którym umieszczone zostaje ciało procesu (programu). W systemie tym kod jest generowany przez kompilatory jako czysta procedura, a zatem może być wielodostępny. Przy powoływaniu następnego procesu wykonywanego na tym samym kodzie (np. ten sam program wywołany przez drugiego użytkownika) dzieli on ten sam segment kodu.

Drugim segmentem jest tzw. segment danych, przeznaczony na dane statyczne procesu. Jest on już indywidualny dla procesu. Trzecim segmentem jest segment stosu, przeznaczony na wektory aktywacji otwieranych dynamicznie bloków.

Przestrzeń adresowa procesu składa się z dwóch obszarów. Pierwszy z nich rozpoczyna się od adresu 0 lokalnego dla procesu i zawiera kod, a ponad kodem - segment danych. Obszar ten może być zwiększany i zmniejszany na żądanie procesu. Drugi obszar zaczyna się od największego adresu możliwego w danej instalacji komputerowej i zawiera segment stosu, rosnący 'w dół'. Obszar ten jest zwiększany automatycznie w wyniku przerwania 'brak pamięci'.

W systemie tym nie ma możliwości 'zaczepienia' procesu w pamięci, tj. wymuszenia, aby proces nie podlegał wymianom. W tym sensie UNIX nie jest systemem czasu rzeczywistego, ponieważ nie może zagwarantować, że czas reakcji nie przekroczy pewnej określonej wartości.

CRQOK: Proces otrzymuje jeden spójny obszar pamięci. Przestrzeń adresowa tego obszaru rozpoczyna się od adresu 0 lokalnego dla procesu. Sposób zagospodarowania pamięci jest pozostawiony procesowi. Przydzielony obszar może być zwiększany, lub zmniejszany na żądanie procesu.

Proces można opatrzyć atrybutem 'rezydentny', wówczas nie będzie on podlegał wymianom. Jest to własność istotna w systemach czasu rzeczywistego.

Obraz procesu w pamięci masowej.

UNIX: Obraz procesu obejmuje segment danych i segment stosu. W chwili, gdy proces podlega wymianie, jego obraz zostaje przechowany w służbowej części nośnika, nie objętej systemem zbiorów. Dotarcie do tego obszaru jest niemożliwe dla użytkowników i programów nie systemowych. Wybór takiego rozwiązania wynikał z dążenia do przyspieszenia operacji wymiany, bowiem obraz procesu może tu być przechowywany w jednym spójnym obszarze nośnika. Natomiast pewne wynikające stąd niewygody można w pewnym stopniu obejść, korzystając z niektórych mechanizmów obsługi przerwań awaryjnych, ich omówienie wykracza jednak poza ramy niniejszego opracowania.

CRQOK: Obraz procesu obejmuje całą pamięć procesu. W czasie operacji wymiany zostaje on przechowany w zbiorze pewnego wyróżnionego typu. Zbiór ten jest tworzony przez system w momencie powoływania procesu. Ponieważ zbiór w tym systemie znajduje się zawsze w spójnym obszarze nośnika, sprawność operacji wymiany przy takim rozwiązaniu jest dobra, zaś użytkownik ma możliwość sięgania do obrazu procesu.

Tablica stanu procesu.

UNIX: Tablica procesu składa się z dwóch części. Pierwsza z nich zawiera tzw. dane systemowe procesu, to jest informacje o procesie, które musi znać system operacyjny w czasie, gdy proces się wykonuje. Są one alokowane przez system w obszarze pamięci procesu. Druga część to informacje o procesie potrzebne w czasie, gdy proces jest 'wypchnięty' z pamięci i jego obraz znajduje się na dysku. Dane te są przechowywane w pamięci alokowanej z zasobów systemowych.

CRQOK: Tablica procesu jest jednoczęściowa, przechowywana w obszarze pamięci alokowanym z zasobów systemowych.

Środowisko procesu.

Środowisko procesu jest tworzone przez jego przodka i składa się ze zmiennych, których wartości mogą wpływać na sposób wykonywania powoływanego procesu. Zazwyczaj przodkiem tym jest język komunikacji systemu z użytkownikiem. Dla uproszczenia mechanizm przekazywania informacji o środowisku zilustrujemy przykładem programu konwersacyjnego, który wyprowadza na końcówkę (monitor) większe ilości informacji. Informacja o środowisku dla takiego programu musi zawierać wskazanie, czy program ma się zatrzymywać po zapełnieniu ekranu a wznowiać pracę na żądanie, oraz ile wierszy mieści ekran.

UNIX: Definicje zmiennych tworzących środowisko przechowywane są przez język komunikacji z użytkownikiem w postaci tekstowej. W omawianym przykładzie informacja o ekranie może być tekstem "window=n", gdzie n jest liczbą wierszy na ekranie, lub zerem gdy wyprowadzanie ma się nie zatrzymywać. Tablica takich tekstów jest kopiowana do pamięci wołanego procesu, zaś odsyłacz do tej tablicy jest przekazywany do procesu jako jeden z argumentów wołania. Proces może przeglądać tablicę tekstów i zanalizować definicję zmiennej "window".

CRQOK: Informacje o środowisku są przekazywane do procesu w dwóch rejestrach, wobec czego są ograniczone do wskaźników bitowych, lub co najwyżej pewnych pól bitowych. Zbiór zmiennych środowiska jest dla danego języka komunikacji ustalony, zaś do procesu przekazywane są tylko ich wartości. Co do naszego przykładu, do procesu zostanie przekazany określony bit w rejestrze, mówiący, czy wyprowadzanie ma być stopowane. Ilość wierszy na ekranie bywa 'zaszyta' w programach.

Jak widać, przekazywanie informacji o środowisku jest bogatsze i bardziej elastyczne w systemie UNIX, ale wymaga od procesu, aby sam analizował definicje zmiennych i dokonywał konwersji z postaci tekstowej.

3.2. Procesy spokrewnione i współpracujące.

Procesami spokrewnionymi będziemy nazywać tu procesy pozostające w relacji przodek-potomek, lub procesy posiadające wspólnego przodka. Procesami współpracującymi zaś - procesy komunikujące się ze sobą w jakikolwiek sposób. Sposób tworzenia procesów spokrewnionych i współpracujących oraz metoda synchronizacji i komunikacji pomiędzy procesami należą do najistotniejszych własności każdego systemu operacyjnego.

Uruchomienie systemu.

W obu omawianych systemach zasada uruchomienia procesów jest taka sama. Bootstrap systemu powoduje utworzenie procesu inicjującego, który jest przodkiem wszystkich innych procesów. Proces inicjujący ustala datę i czas oraz aktualną konfigurację sprzętu, po czym powołuje procesy dla obsługi końcówek.

W systemie UNIX dla każdej końcówki utworzony zostaje osobny proces, zaś w systemie CROOK - jeden proces wielodostępny tworzony jest dla grupy kilku końcówek. Proces obsługi końcówki jest to zazwyczaj interpreter zleceń pisanych przez użytkownika. Interpreter ten rozpoznaje zlecenie, po czym powołuje odpowiedni proces dla jego wykonania. Każdy proces może powoływać własnych potomków, tworząc w ten sposób dynamicznie zmieniające się drzewo procesów.

Mechanizmy synchronizacji.

W systemach operacyjnych do najczęściej używanych mechanizmów synchronizacji należą semaforey, przerywania, wspólne (dzielone) zmienne i obszary pamięci, mechanizmy zdarzeń i przesyłanie komunikatów.

UNIX: System ten nie posiada semaforów, komunikatów, ani zmiennych wspólnych dla procesów. Brak tych narzędzi jest od kilku lat przedmiotem krytyki. Podstawowymi mechanizmami synchronizacji są zdarzenia i przerywania.

Mechanizm zdarzeń polega na tym, że proces może zażądać (od systemu) wysłania określonego sygnału do wszystkich procesów, które na taki sygnał oczekują. System przewiduje 256 sygnałów, rozróżnianych numerami. Istnieje też ekstrakod, za pomocą którego proces może zadeklarować, że czeka na określony sygnał. Sygnały nie są zmiennymi, nie posiadają wartości i nie są przez system pamiętane. Toteż w przypadku, gdy sygnał zostanie wysłany wcześniej, niż jakkolwiek proces zasygnalizuje, że na ten sygnał czeka, jest on gubiony.

Niektóre przerwania mogą być używane do synchronizacji. Proces ma do dyspozycji ekstrakody, za pomocą którego deklaruje własną obsługę określonych przerw. Przerwania takie nie są wówczas obsługiwane przez system, lecz przekazywane do procesu.

Chociaż narzędzia te sprawiają wrażenie ubogich, należy jednak zaznaczyć, że za wyjątkiem zastosowań w systemach czasu rzeczywistego, są na ogół wystarczające. Należy też wspomnieć, że semaforów można tworzyć na poziomie użytkowym za pomocą zbiorów. Korzysta się tu z faktu, że tworzenie zbioru jest w systemie operacją spójną.

CROOK: Mechanizm 'przechwytywania' przerw przez proces jest tu podobny, jak w systemie UNIX. Mechanizm zdarzeń natomiast znacznie się różni. Zdarzenia, nazywane tu alarmami, są adresowane (do przodka) i są przez system kolejgowane osobno dla każdego adresata. Proces zawieszony zostaje w wyniku adresowanego doń alarmu wznowiony, zaś proces aktywny może zawsze dowiedzieć się o alarmy niezależnie od czasu, w jakim wystąpiły.

System wyposażony jest w ekstrakody semaforowe i ekstrakody do przesyłania komunikatów między procesami. Dla procesów blisko spokrewnionych możliwe jest także tworzenie dzielonych obszarów pamięci.

Komunikacja.

Podstawowym mechanizmem komunikacji procesu z otoczeniem w obydwu systemach są tzw. strumienie. Strumień jest obiektem umownym, łączącym źródło i odbiorcę informacji. W typowym przypadku jednym z nich jest proces, zaś drugim - zbiór lub urządzenie. Implementacja strumienia w systemie wiąże się z utworzeniem odpowiedniego bufora i użyciem odpowiednich mechanizmów synchronizacji, powodujących zawieszanie nadawcy (odbiorcy) odpowiednio do stanu bufora.

Komunikacja pomiędzy procesami jest zorganizowana odmiennie. W systemie UNIX używany bywa do tego celu specjalny rodzaj strumienia, tzw. 'pipe', łączący ze sobą dwa procesy. W systemie CROOK, podobnie jak w przypadku synchronizacji, używany jest mechanizm przesyłania komunikatów, a w przypadku procesów blisko spokrewnionych - dzielone obszary pamięci.

Poza tym możliwe są oczywiście wszystkie typowe sposoby przekazywania informacji, np. poprzez wspólne zbiory.

W obydwu systemach zarządzanie procesami jest zorganizowane w elegancki i elastyczny sposób. W systemie CROOK nieco bogatsze są mechanizmy synchronizacji, zaś silną stroną systemu UNIX jest mechanizm 'pipe', umożliwiający łączenie zleceń i programów w potok przetwarzający.

4. Podsumowanie

System UNIX znajduje się w użytkowaniu już dwanaście lat, natomiast CROOK-4 trzeci rok. Naturalnie więc UNIX jest w większym stopniu 'odplukwiony' (choć nie pozbawiony całkowicie błędów) i zdążył 'obrosnąć' ogromną ilością bibliotek, pakietów aplikacyjnych i rozmaitych narzędzi programowych.

Natomiast CROOK-4 powstawał znacznie później, w czasie, gdy słabe strony systemu UNIX były już ogólnie znane (wskazywane zresztą uczciwie przez samych jego autorów). Toteż zawiera on pewne elementy, których brakuje w systemie UNIX. Jednakże musi upłynąć jeszcze sporo czasu, zanim doczeka się równie bogatego oprogramowania użytkowego.

Dokonana tu pobieżna analiza pokazała, że pod względem własności użytkowych i funkcjonalnych są one bardzo podobne. Interesujące natomiast jest obserwowanie, jak w obydwu tych systemach równoważne cele osiągnane są czasami zupełnie różnymi środkami. Jeżeli niniejsze opracowanie zachęci czytelników do bliższego zapoznania się z systemami UNIX i CROOK, to jego cel można będzie uważać za osiągnięty.

mgr inż. Jan Wierzbicki
mgr inż. Maria Perek
Huta Szkła Okiennego
"Szczakowa"

PROBLEMY PRZENOSZENIA OPROGRAMOWANIA
UŻYTKOWEGO Z SYSTEMU SOM-3 NA CROOK-4

Ośrodek Informatyki w Hucie Szkła Okiennego "Szczakowa" jako jeden z nielicznych zdecydował się na przejście z 2-zadaniowego SOM'a-3 na CROOK'a-4.

Dysponujemy obecnie dwoma systemami MERA-400 w łącznej konfiguracji 0.8 Mbajta pamięci operacyjnej, 4 dyski tj. 20 Mbajtów / w niedługim czasie 6 dysków typu Winchester /. Ponieważ na jednym systemie eksploatujemy bogate oprogramowanie użytkowe, drugi system pozwolił nam na rozpoczęcie prac nad przejściem na CROOK'a-4.

Różnice między systemem operacyjnym CROOK-4 i SOM-3 są znaczne. Do najistotniejszych należą: dynamiczne przydzielanie pamięci operacyjnej zadaniom oraz znacznie bogatszy i wygodniejszy w użytkowaniu system zbiorów dyskowych pod systemem CROOK-4. System CROOK-4 posiada odmienny zestaw dyrektyw a także własny zestaw ekstrakodów. Ponieważ, ogólnie biorąc, możliwości systemu operacyjnego CROOK-4 są bogatsze od możliwości systemu SOM-3, przenoszenie programów działających pod systemem SOM-3 jest na ogół możliwe bez zasadniczych zmian w organizacji oprogramowania. Dla pojęć i funkcji systemu SOM-3 można bowiem znaleźć odpowiednik w systemie CROOK-4. W szczególności pojęciu sekcji dyskowej w systemie SOM-3 można przyporządkować pojęcie zbioru dyskowego w systemie CROOK-4. Większość ekstrakodów systemu SOM-3 posiada swoje odpowiedniki w systemie CROOK-4.

Decydując się na tak poważne przedsięwzięcie jak przejście na CROOK'a zdawaliśmy sobie sprawę z trudności, jakie możemy napotkać zwłaszcza, że nasze oprogramowanie opiera się na bazie danych BD-83. Pierwszym etapem, który został już zakończony było właśnie przerebienie bazy danych BD-83 Vitrin. Kolejne etapy to kompleksowe przetwarzanie poszczególnych systemów użytkowych i rozpoczynanie eksploatacji pod CROOK'iem. Chcielibyśmy przedstawić Państwu, a w szczególności Ośrodkom, które będą chciały przejść na CROOK'a, pewne problemy z jakimi będą mogli się spotkać w trakcie przerabiania.

Ponieważ całe nasze oprogramowanie użytkowe jest napisane w języku Fortran więc tylko na ten temat możemy się wypowiadać. Programy, które eksploatujemy w głównej mierze dotyczą przetwarzania jak również w niewielkim stopniu obliczeń inżynierskich. Ponieważ kilka Ośrodków o podobnym profilu działalności zamierza przejść na CROOK'a uważaliśmy za celowe zabranie głosu w tej sprawie.

Większość zmian jakie trzeba wprowadzić w programach fortranowskich wynika przede wszystkim z różnych wersji kompilatorów Fortranu w obu systemach. Jakkolwiek w oprogramowaniu systemu CROOK-4 istnieje program FORSC, który zmienia wersję źródłową programu fortranowskiego w standardzie SOM-3 na standard CROOK-4, to jednak nie można zalecić stosowania tego programu, ponieważ w wyniku jego działania można uzyskać niepoprawny tekst, który będzie znacznie trudniej poprawić niż tekst wyjściowy.

Ponadto program ten wykonuje jedynie część potrzebnych poprawek, które można stosunkowo łatwo a napewno znacznie bezpieczniej wykonać edytorem tekstowym EXM stosując operacje globalne obejmujące cały tekst poprawianego programu.

Swego czasu były robione ustalenia co do programu, który w solidny sposób przekładałby programy fortranowskie z wstawkami assemblerowymi z SOM'a na CROOK'a.

Niestety prawdopodobnie skończyło się tylko na ustaleniach.

Oceniając obecną wersję kompilatora programów fortranowskich należy powiedzieć, że nie możemy mieć do niej większych zastrzeżeń. Programy stosunkowo duże, po 1000-2000 instrukcji z 3,4 poziomami nakładek pracują prawidłowo. Należy tutaj szczególnie podkreślić szybkość translacji, wygodnie rozwiązany problem nakładkowania, a szczególnie dobrze rozwiązaną komunikację o błędach.

Działanie programów fortranowskich pod CROOK'iem-4 na zbiorach bezpośredniego dostępu jest jednak znacznie wolniejsze od analogicznych pod SOM'em. Przyczyną jest prawdopodobnie złe sformułowanie instrukcji czytania i pisania z/do zbiorów, lecz mamy nadzieję, że prace w tym kierunku nie zostały jeszcze zakończone.

Najwięcej problemów sprawiło nam wykrycie przyczyny pojawiających się błędów END OF FILE /KONIEC ZBIORU/, które często wy-

stępowwały podczas translacji naszych programów. Dzięki pomocy Autorów systemu CROCK-4 udało się ustalić, że przyczyną pojawiania się błędów jest niedopuszczalny sposób użycia instrukcji DATA dla zmiennych występujących w instrukcji COMMON. DATA można nadać wartości zmiennym z obszaru wspólnego tylko w pierwszym segmencie, w którym użyto instrukcji COMMON dla tych zmiennych.

Brak programu tłumaczącego, drobne błędy w Fortranie oraz przedstawione błędy w kompilacji nie są istotną przeszkodą w przerabianiu oprogramowania. Wprawny programista znający różnice między kompilatorami jest w stanie przerobić program około 1000 linii w ciągu 2-3 dni, a więc przerabianie tego typu programów nie jest rzeczą zbyt trudną. Istotną sprawą jest również zaprojektowanie struktury użytkowników, zbiorów i kaset.

Aby skopiować program źródłowy w Fortranie z SCM'a na CROCK'a można:

- wczytać tasiemkę wyperforowaną w reżimie NOC /co proponują autorzy Fortranu/,
- wykorzystać symulator SOM-3 i dyrektywę COPY, co jest, znacznie wygodniejsze gdyż nie wymaga przegrywania programów na tasiemki.

Na zakończenie przedstawimy podstawowe zmiany jakie należy wprowadzić w programach fortranowskich aby mogły pracować pod CROCK`iem:

1. Program należy uzupełnić wierszami sterującymi kompilacją np. #FOR, #LIB itp.,
2. Nazwę PROGRAM należy zastąpić nazwą MASTER
3. W Fortranie CROCK nie można stosować dyrektywy IMPLICIT
4. Fortran CROCK umożliwia pisanie wstawek assemblerowych podobnie jak pod SCM`em jednak sposób zapisu jest całkowicie odmienny
5. Znakem kontynuacji może być jedynie znak "&", a nie dowolny znak w kolumnie 5
6. Znak "&" jest używany zamiast znaku "D" w pierwszej kolumnie dla oznaczenia rozkazów kompilowanych opcjonalnie
7. W logicznych instrukcjach IF nie należy stosować etykiet
8. W instrukcjach WRITE i READ używa się numerów strumieni od 0 do 20 i od 30 do 50
9. Instrukcje INREC i OUTREC zamienić na READ i WRITE

10. Zbiory bezpośredniego dostępu są dostępne za pomocą instrukcji READ, WRITE - możliwe jest to po użyciu instrukcji SETFILE
11. Zamienić instrukcje ENCODE i DECODE na READ i WRITE
12. Poprawić znaki sterujące wydrukiem
13. Wskaźniki w Instrukcji DO nie mogą być wyrażeniami
14. W instrukcji FORMAT nie występuje format typu C dla czytania i pisania danych w CAN-kodzie
15. Nie dopuszcza się stosowania instrukcji wielokrotnego podstawienia
16. Nie stosuje się instrukcji PAUSE i STOP z łańcuchami alfanumerycznymi.

inż. Andrzej Pisiewicz
Wojewódzkie Biuro Projektów w Zabrze
ul. Wolności 186 41-800 Z a b r z e
tel. 71-20-21 w. 84

DOŚWIADCZENIA Z EKSPLOATACJI SYSTEMÓW
OPERACYJNYCH CROOK I DSM - 1

Nawiązany z początkiem 1981 roku kontakt z P.P. MORS w Gdyni, artykuł w miesięczniku Informatyka 6/80 skłoniły nas do zainteresowania się systemem operacyjnym CROOK-3. Posiadanie w swych zasobach systemowych kompilatora FORTRAN'u oraz symulatora SOM-3 skłoniło nas w 1982 roku do kupna tego systemu. Za naszym przykładem poszło jeszcze kilka przedsiębiorstw na Śląsku.

Dostarczona wersja systemu nie posiadała w swych zasobach kompilatora FORTRAN'u, ale wg zapewnień dystrybutora jakim był w tym okresie MORS braki miały być w niedalekiej przyszłości uzupełnione. Z systemem CROOK-3 wiązaliśmy wielkie nadzieje i po rozbudowie sprzętowej MERY /w tym okresie był to standard doposażony w 3 monitory MERA 7952 VGD/ miał on zrewolucjonizować pracę w Biurze. Pierwsze prace pod nowym systemem dawały dużo satysfakcji. Łatwiejsza obsługa, szybkość działania przejrzystość komunikatów podawanych przez system była nieporównywalna z dotychczasową pracą.

Ponieważ oprogramowanie, które posiadaliśmy było w większości napisane w Fortranie, a i sami tworzyliśmy je w tym języku, dlatego też interpreter języka BASIC /nieporównywalny z dostarczonym przez producenta/ był niewystarczający. Również niezbędny był sprawny symulator SOM-3, gdyż posiadaliśmy już dość bogatą bibliotekę programów zakupionych w innych ośrodkach, a nowe oprogramowanie powstawało w kraju dalej pod systemem SOM-3.

Niestety testowane coraz to nowsze wersje symulatora nie dawały pozytywnych rezultatów. Przy okazji, w trakcie prac wykrywaliśmy błędy istniejące w samym systemie operacyjnym.

Taki stan trwał około roku. Byliśmy dość odległą placówką doświadczalną starającą się eksploatować zachwalany /bezsposornie słusznie/ system operacyjny CROOK-3 lecz bez zadawalających nas rezultatów.

Niesprawny symulator SOM-3 jak i obiecowany i nie dostarczony kompilator Fortranu nie zezwoliły nam na całkowite przejście pracy ośrodka pod system CROOK-3.

Pierwsza konferencja użytkowników mini komputera MERA 400 w Gdańsku 1983 roku potwierdziła potrzebę stworzenia i rozpropagowania nowego, sprawniejszego systemu operacyjnego. Systemu wieloprogramowego i wielozadaniowego. Był nim CROOK-3, właściwie już wtedy jego następcą CROOK-4. Na tej konferencji pojawiła się informacja o kompilatorze języka FORTNAN - CROOK, ale dostępnego tylko pod systemem CROOK-4.

Ważnie jakie wystąpiły pomiędzy dystrybutorem /MORS/, a autorem /Politechnika Gdańska/ odcięły nas od możliwości zainstalowania tego języka w naszym Biurze, gdyż systemy CROOK-3 i CROOK-4 różniły się od siebie.

Pomni przykrych doświadczeń nie chcieliśmy kupować w ciemno nowego systemu. Dzięki uprzejmości nowego dystrybutora /Instytut Maszyn Matematycznych w Warszawie/ uzyskaliśmy zgodę na okresową dzierżawę systemu CROOK-4, wraz z kompilatorem FORTRAN-CROOK i symulatorem SOM-3.

Od września 1984 roku /po przeszkoleniu w tzw. szkole Crook'a staliśmy się znów placówką doświadczalno-testującą system operacyjny, kompilator Fortranu oraz symulator SOM-3. Przetestowane zostały u nas wersje systemu od DB 49/13 do DB/56/1, kompilator języka FORTRAN-CROOK od wersji 26/3 do 30/0, oraz symulator SOM-3.

Wykrywane błędy lub nieprawidłowości działania systemu operacyjnego były sukcesywnie usuwane, tak że ostatnia

wersja jaką posiadaliśmy /DB/56/1/ wg naszej oceny była bez błędów.

Negatywnie ocenić musimy działanie symulatora SOM-3. Wprawdzie ostatnia, testowana wersja z lutego br. miała już mniej błędów niż wiele poprzednich, lecz nieuzasadnione spowalnianie transmisji na ekran monitora, możliwość pracy tylko jednego użytkownika pod symulatorem, abortowanie niektórych programów fortranowskich korzystających z bezpośredniego dostępu, to błędy wykluczające możliwość pracy pod tym symulatorem.

Również FORTRAN-CROOK nie spełnił podkładanych w min nadziei. W pierwszych wersjach testowanych przez nas były błędy i niezgodności działania kompilatora zgodnie z dostarczoną dokumentacją.

Były one niwelowane w miarę pojawiania się nowych wersji.

Byliśmy przekonani i dalej stoimy na takim stanowisku, iż kompilator FORTRAN-CROOK nie powinien być uboższy, a conajmniej równy FORTRAN-owi IV S dlatego, że powstawał dla MERY 400, /a nie innej maszyny/ a ta w tym czasie miała już bogate oprogramowanie właśnie napisane w tym języku. Okazało się, że nawet posiadając źródła programów nie można było przenieść oprogramowania bez ingerencji w źródło lub wręcz zmianie musiałyby ulec koncepcja rozwiązania. Dotyczyło to np. całego pakietu programów konwersacyjnych wykonanych w naszym Biurze dla systemu PŁACE.

Z wyżej wymienionych powodów zmuszeni byliśmy do zrezygnowania z systemów rodziny CROOK, ale potrzeba instalacji systemu o podobnych parametrach była nieunikniona. Zaczęliśmy rozglądać się za systemem, który w pełni akceptowałyby programy SOM-owskie, był wielodostępny wieloprogramowy i pracował niezawodnie.

Po przeanalizowaniu możliwości systemów autorstwa Instytutów Informatyki w Krakowie i Warszawie oraz Instytut Ciężkiej Syntezy Organicznej "Błachownia" wybraliśmy ten ostatni.

Przemawiał za nim brak sztywnych przydziałów sekcji

roboczych, brak wydzielonej konsoli operatora, zmienny przydział pamięci operacyjnej dla zadania, pełna akceptowalność programów i zbiorów SOM-owskich, automatycznie rozszerzanie sekcji do rozmiaru zbioru /podobnie jak w CROOK4/ oraz bliskość ośrodka autorskiego.

W kwietniu 1984 roku zainstalowaliśmy na MERA-400 o PAO 160kB, dwóch stacjach dyskowych MERA 7925, trzech monitorach MERA 7952 VGD i dwóch drukarkach, system operacyjny DSM-1. Ponieważ system ten akceptuje kasety dyskowe z różną organizacją, również organizacją FMC, pierwsze tygodnie pracy odbywały się na kasetach wymiennych z organizacją SOM/FMC. Bardzo różnorodna ilość procedur przechowywana w bibliotece słownikowej, jak również napisane własne procedury do celów uruchomienia programów aplikacyjnych pozwoliły na obsługę istniejącego oprogramowania w sposób identyczny jak pod SOM-3. A różnica nie była zauważalna nawet przez operatorów EMC.

Sukcesywnie oprogramowanie było przenoszone pod nową organizację, zapewniającą zmniejszenie obszarów zajmowanych przez binaria programów, oraz tworzone były nowe kasety wymienne, głównie dla celów projektantów ETO, oraz przechowywania zbiorów.

Oczywiście nie obyło się bez kłopotów, ale bliskość ośrodka autorskiego, oraz dogłębna znajomość zagadnienia przez autora systemu, pozwoliły na szybkie usuwanie błędów jak również dostosowanie systemu do naszych, wewnętrznych potrzeb oraz wymogów.

Z dużych, powszechnie eksploatowanych w Biurach Projektów programów pracują poprawnie PROBUS, WB21, WB22 i inne, a co najważniejsze nasz cały pakiet programów PŁACE, którego w żaden sposób nie można było uruchomić ani pod symulatorem SOM-3, ani kompilatorem XFOR w systemie CROOK-4.

Wnioski

Należy żałować, że doskonały w założeniach, funkcjonalny, bardzo szybki system operacyjny CROOK nie został od razu wyposażony w pełni sprawny symulator systemu SOM-3 jak i kompilator FORTRAN'u nie gorszy niż ten pracujący pod systemem SOM-3/FMC.

Należało się spodziewać, że jednostki eksploatujące oprogramowanie użytkowe nie mogą sobie pozwolić przy zmianie systemu operacyjnego na przerwę w normalnej eksploatacji.

Tej przerwy nie mieliśmy przechodząc pod system DSM-1. Jest on wprawdzie wolniejszy od CROOK'a, Należy go na nowo generować przy zmianie /rozbudowie/ konfiguracji EMC, gdzie w CROOK'u wystarczy zmiana w tablicy konfiguracyjnej.

Jest somo-podobny, ale rozbudowany aparat makrodyrektyw umożliwiający ich dowolne zagłębianie czyni pracę podobną do pracy pod CROOK'iem. Brak potrzeby pisania dodatkowych znaków przy nasłuchu dyrektyw JOB'a /g lub g Do/.

Szybki MAKROASSEMBLER, oraz LINK EDITOR dający porządne komunikaty wykrywanych błędów, a także sprawny podsystem zbiorów będący pod kontrolę systemu operacyjnego, a dający swobodę w tworzeniu zbiorów to między innymi walory systemu DSM-1, który mogę polecić z czystym sumieniem jednostkom o bogatym już oprogramowaniu somowskim.

Instytut Ciężkiej Syntezy Organicznej "Młochowia"
47-232 Kędzierzyn-Koźle
ul. Energetyków 9

Pracownia Matematyki Stosowanej
mgr inż. Henryk Olkowski tel. 332-41 w. 5459

System operacyjny DSM-1

1. Ogólna charakterystyka systemu.

System operacyjny DSM-1 przeznaczony jest przede wszystkim dla użytkowników MERY-400 pracujących dotychczas pod systemem operacyjnym SOM-3.

Spełnia on praktycznie wszystkie funkcje systemu SOM-3, pozwalając wykorzystywać procesory tego systemu oraz opracowane dotychczas oprogramowanie użytkowe, zarówno w postaci binarnej jak i źródłowej.

W ośrodku autorskim wykorzystuje się przykładowo pod kontrolą systemu DS M-1 translator FORTRAN-u IV (działający dwukrotnie szybciej - dzięki szybszemu translatorowi MACROASSEMBLER-a) i interpreter BASIC.

Wykorzystywano również standardowy translator MACROASSEMBLER-a, konsolidator EDI oraz bibliotekarz CAT.

Obecnie procesorów tych się nie wykorzystuje, ponieważ ich funkcje przejęły opracowane procesory systemu DSM-1 o znacznie większych możliwościach.

W systemie DSM-1 brak jest wielu, szczególnie dotkliwych w systemie SOM-3 ograniczeń.

Możliwa jest przykładowo jednoczesna praca wielu zadań z dynamicznym przydziałem pamięci operacyjnej i bardzo elastyczne oraz oszczędne wykorzystanie pamięci dyskowych dzięki wbudowanemu w system DSM-1 podsystemowi zbiorów.

Należy wyraźnie zaznaczyć, że system DSM-1 nie jest "poprawionym" systemem SOM-3, ale zupełnie nowym systemem opracowanym w ten sposób, aby obsłużyć przez niego zadania jak najbardziej przypominające działanie systemu SOM-3.

2. Zadania.

Pod kontrolą systemu DSM-1 może pracować jednocześnie wiele zadań. Przydział pamięci operacyjnej dla zadań jest dynamiczny.

Każde zadanie może w trakcie działania określać potrzebną dla niego wielkość pamięci operacyjnej.

W przypadku, gdy zadania zadania przekraczają wielkość dostępnej wolnej pamięci, uruchamiane są automatycznie przez system wymiany pamiędzi

pamięci operacyjnej i dyskowej, pozwalające na zwolnienie obszaru pamięci potrzebnego dla aktualnie obsłużonego zadania.

System stosuje strategię przydziału maszyny zadaniom oparte o wykorzystanie przez nie czasu maszyny.

Pozwala ona w przypadku zadań o różnych priorytetach na równomierne wykorzystanie maszyny przez wiele zadań.

Jeśli ciało zadania, które ma być unktuwnione zapisane jest aktualnie na dysku, to wprowadzane jest ono do pamięci operacyjnej (po ewentualnym przesłaniu na dysk, w razie potrzeby, ciała innych zadań).

Umożliwia to użytkownikom bardzo wygodne korzystanie z systemu.

Przykładowo można w zadaniach wprowadzać dane z klawiatury, nie odczuwając praktycznie dodatkowego wykorzystywania maszyny przez inne wykonujące obliczenia zadania.

Opisany sposób zapewnia możliwość jednoczesnej pracy wielu użytkowników, nawet przy braku dostatecznie dużej pamięci operacyjnej.

Przy minimalnej wielkości pamięci operacyjnej (32 k) system zajmuje 12 k, pozostawiając do dyspozycji zadania obszar 20 k.

Przy większej pamięci operacyjnej (lub ograniczeniu wielkości pamięci dla zadań do 16 k), możliwa jest jednoczesna praca większej ilości zadań.

Każde zadanie ma w systemie swój opis zawierający między innymi :

- definicje zbioru bazowego i roboczego zadania (określające wykorzystywane przez nie fragmenty struktury zbiorów na plikach dyskowych).
- Potrzebną początkową wielkość pamięci operacyjnej.
- Nazwy strumieni stałych oraz określenie ich początkowych standardowych przydziałów (urządzenia, strumienie, sekcje ze zbioru bazowego lub sekcje robocze).

Informacje te mogą być zmieniane przy pomocy dyrektyw zadania komunikacji, pozwalając przykładowo na zmianę wykorzystywanych przez zadanie urządzeń czy definiowanie potrzebnych strumieni we/wy.

Wymagane jest tylko w zadaniach występowanie strumieni CI oraz CO, nazwy pozostałych strumieni i ich ilość mogą być praktycznie dowolne.

Opisy zadań umieszczone są w specjalnym katalogu systemowym.

Identyfikatorem zadania w systemie jest trzycyfrowa nazwa.

Można zdefiniować dla zadania hasło ochronne, bez znajomości którego nie można zmienić opisu zadania ani go uruchomić.

Uruchomienie zadania odbywa się standardowo przy pomocy dyrektywy Zadania Komunikacji.

Po pomyslnym przydziale wszystkich potrzebnych do startu zadania zasobów, rozpoczyna działanie procesor JOB (umożliwiająca użytkownikowi rozpoczęcie pracy).

Uruchomione zadanie może korzystać z własnych (lub przydzielonych

mu do wyłącznego korzystania przez system) sekcji roboczych.

Zadania mogą wykorzystywać informacje lokalne jak i globalne (wspólne dla wszystkich zadań).

Każdy użytkownik ma wrazenie wyłączości korzystania z maszyny; zwłaszcza w przypadku pracy z końcówki umożliwiające wysyłanie przerwania ZGŁOSZENIE OPERATORA; powodujące zyskanie się wielodostępnego Zadania Komunikacji systemu DSM-1 bezpośrednio na końcówce.

3. Rozszerzenia podsystemu we/wy.

Operacje we/wy w zadaniu mogą być wykonywane za pośrednictwem strumieni oraz tzw. numerów logicznych.

Numery logiczne (standardowo oznaczane liczbami od 1 do 20) są specjalnymi kanałami dostępu do sekcji dyskowych.

Można przy ich pomocy, w prosty sposób, wykonywać szereg operacji typowych dla tworzenia i aktualizacji baz danych :

- tworzyć i kasować sekcje dyskowe;
- odczytywać i zmieniać informacje o sekcjach (np. ilość i wielkość rekordów),
- odczytywać, zapisywać, wstawiać, usuwać oraz zamieniać dowolne grupy rekordów sekcji.

Przesyłane za pośrednictwem numerów logicznych treści rekordów nie podlegają żadnej interpretacji.

Operacje przesłania realizowane są przy pomocy ekstrakodów i wymagają jedynie podania : numeru logicznego; numeru początkowego rekordu; ilości rekordów oraz adresu tablicy na treść rekordów przy odczycie (lub z treścią rekordów przy zapisie).

Wiele ekstrakodów umożliwia operacje na buforach we/wy.

Bufory posiadają opis określający ich wielkość; ilość zawartych w nich znaków; wskaźnik bajtowy sterujący odczycem z bufora oraz adres tablicy UFT identyfikującej strumień we/wy.

Dzięki temu operacje na buforach są bardzo proste; jedynym parametrem który należy podawać jest adres bufora.

Bufory wykorzystywane są przez wiele ekstrakodów wydruku oraz odczytu i rozpoznawania parametrów.

Istnieje również opcjonalna możliwość wykorzystania buforów przez ekstrakody READ i WRITE.

Przy operacjach we/wy wykonywanych za pośrednictwem strumieni istnieje możliwość rozszerzenia tablicy UFT pozwalającego definiować adres procedury obsługi błędów i zdarzeń operacji we/wy.

Upraszcza to bardzo obsługę błędów operacji oraz operacji wykonywanych w trybie QUICK-RETURN.

Operacje we/wy na niektórych urządzeniach (szczególnie wczytywania

z klawiatury) mogą być buforowane przez system.

Pozwala to na wykorzystanie, w czasie trwania operacji, obszaru pamięci operacyjnej zajętego przez ciało zadania przez inne zadania.

4. Ekstrakody systemu DSM-1.

System posiada rozszerzone przeszło dwukrotnie listę ekstrakodów.

Oprócz prawie wszystkich ekstrakodów systemu SOM-3 (za wyjątkiem operacji na sygnałach) system DSM-1 posiada dodatkowe ekstrakody.

Umożliwiają one między innymi :

- otwieranie dostępu do zbiorów dyskowych.
- Uzyskiwanie informacji o zbiorach i sekcjach w zbiorze.
- Tworzenie oraz kasowanie sekcji dyskowych.
- Otwieranie dostępu do sekcji dyskowych (poprzez ich przydziały do strumieni we/wy i numerów logicznych).
- Aktualizację informacji na sekcjach (i o sekcjach) za pośrednictwem numerów logicznych.
- Wydruki informacji (liczb całkowitych w różnych postaciach, tekstów oraz spacji) do buforów oraz wyprowadzanie zawartości buforów na strumienie we/wy.
- Odczyt informacji z buforów.
- Kodowanie i rozkodowywanie dat (kod daty jest liczbą całkowitą i określa różnicę w dniach pomiędzy podaną datą i datą 50.01.01).
- Generowanie liczb losowych o rozkładzie równomiernym.
- Uzyskiwanie dodatkowych informacji o zadaniu i jego overlay'ach.

W minimalnych wersjach systemu, ciała części rzadko używanych ekstrakodów mogą być umieszczone na dysku.

Wprowadzane są one, w razie potrzeby, automatycznie przez system do obszaru wspólnego dla wszystkich nierezynujących ekstrakodów.

Pozwala to na dodatkowe rozszerzenia listy ekstrakodów bez konieczności zwiększania wielkości systemu.

5. Zadanie Komunikacji.

Zadanie to służy do komunikacji operatora z systemem.

Zsłusza się ono na konsoli operatora bezpośrednio po wczytaniu systemu do pamięci operacyjnej oraz po naciśnięciu przycisku OPRG na pulpicie maszyny.

W rozrzeszonej wersji systemu, wielodostępne Zadanie Komunikacji może również zasłaskać się na dowolnej końcówce, jeżeli istnieje możliwość przesłania z tej końcówki przerwania (ZGROSZENIA OPERATORA).

• Przy pomocy dyrektyw Zadania Komunikacji można :

- wykonywać operacje na systemowym katalogu zadań np. dodawać, usuwać lub poprawiać opisy zadań.
- Uzyskiwać informacje o czasie maszyny zużytych przez poszczególne zadania.
- Uzyskiwać informacje o wykorzystaniu poszczególnych zadań (daty oraz czas zużyty przy każdym uruchomieniu w ciągu pewnego okresu czasu).
- Uruchamiać zadania oraz wyrzucać je z kolejki systemowej.
- Zmieniać informacje w działających zadaniach oraz zatrzymywać je i wznowiać.
- Zmieniać definicje zbiorów zadania oraz wykonywać operacje we/wy za pośrednictwem jeso strumieni.
- Zawieszal i odwieszal działanie urządzeń we/wy.
- Zwalniał i przekazywał systemowi wymienne pakiety dyskowe.
- Uzyskiwał najrozmaitsze informacje o działających zadaniach.
- Uzyskiwał informacje o zasobach systemowych.
- Zapisywał i odtwarzał stan zadań.
- Zmieniał parametry systemowe urządzeń znakowych (np. wielkości rekordów).
- Zatrzymywał i wznowiał pracę systemu.
- Wprowadzał inną wersję systemu.
- Wykonywał szereg innych pomocniczych operacji.

6. Podsystem zbiorów dyskowych.

Wbudowany w system DSM-1 podsystem zbiorów umożliwia korzystanie z sekcji dyskowych, które identyfikowane są poprzez ciąg nazw symbolicznych katalogu nadrzędnego (zbioru) oraz nazwę symboliczną sekcji.

Nazwy nie są związane z fizycznym rozmieszczeniem sekcji na pakiecie dyskowym.

Sekcjom przydzielane są obszary dyskowe składające się z kolejnych sektorów, a wielkość takiego obszaru może zmieniać się od zera do całkowitej ilości sektorów aktualnie dostępnych na pakiecie dyskowym.

Ilość sekcji, które można utworzyć na jednym pakiecie, jest zależna praktycznie tylko od ich łącznej wielkości.

Pozwala to na zastąpienie zapisu wielu, rozdzielonych znacznikami końca zbioru, informacji na jednej sekcji przez zapis tych informacji na oddzielnych sekcjach.

Rozwiązanie takie zapewnia szybszy dostęp do informacji oraz zmniejsza wielkość zajętego obszaru dyskowego.

W podsystemie zbiorów (systemu DSM-1) największym adresowalnym elementem jest pakiet dyskowy; utożsamiany ze zbiorem głównym zapisanym na tym pakiecie.

Zbiór i pakiet mają tę samą nazwę (nazwa pakietu zapisana jest jako klucz w polu adresowym sektorów pakietu).

Nazwy wszystkich pakietów dyskowych używanych pod kontrolą systemu powinny być różne. Zabezpiecza to przed przypadkowym wykonaniem operacji na niewłaściwym pakiecie.

Wielopoziomowa struktura informacji dyskowych można przedstawić następująco :

- na każdym pakiecie dyskowym zapisany jest katalog główny (zbiór główny) o nazwie zgodnej z nazwą pakietu;
- każdemu katalogowi (zbiorowi) mogą podlegać katalogi (zbiory) i sekcje;
- każdy element struktury (katalog lub sekcja), identyfikuje się poprzez nazwę elementu i ciąg nazw katalogów nadrzędnych.

Sekcje dzielą się na rekordy o stałej wielkości.
Rekord sekcji może składać się z 1 do 4095 słów.

Podsystem zbiorów zapewnia :

- tworzenie i kasowanie sekcji przez użytkowników;
- automatycznie rozszerzanie w razie potrzeby obszaru przydzielonego sekcji;
- dostęp do informacji o sekcji zawierających m.in : datę utworzenia i ostatniego zapisu, adres i wielkość obszaru przydzielonego sekcji, ilość i wielkość zapisanych rekordów;
- możliwość ochrony treści sekcji przed zapisem; ochrony sekcji przed skasowaniem i dostępem innych zadań;
- szybki dostęp do każdego elementu struktury zbiorów;
- gospodarke wolnymi obszarami na pakiecie dyskowym (prowadzona jest ich ewidencja) i możliwość ich scalania w razie potrzeby;
- korzystanie z sekcji roboczych kasowanych automatycznie przy zakończeniu działania zadania.

Dzięki wielopoziomowej strukturze informacji możliwe jest tworzenie

sekcji o dowolnych nazwach, niezależnych od nazw sekcji w innych zbiorach, jak i również tworzenie podobnych struktur zbiorów.

Pozwala to przykładowo na wykorzystywanie tych samych makrodyrektyw \$UD procesora JOB do operacji na informacjach wielu zadań.

Użytkownik nie ma dostępu do informacji dyskowych poza podsystemem zbiorów. Zapewnia to ochronę informacji zapisanej na pakietach dyskowych, przed skutkami błędów programowych.

Pod kontrolą systemu operacyjnego DSM-1 odczyt i zapis informacji dyskowych jest możliwy przy wykorzystaniu pakietów zorganizowanych wg. zasad podsystemu zbiorów systemu DSM-1.

Możliwe jest również bezpośrednie wykorzystywanie pakietów dyskowych zorganizowanych wg. innych zasad niż w systemie DSM-1.

Przykładowo: opisywany system pozwala wykorzystywać standardowe pakiety SOMU-3 (o kluczu równym -1), pakiety zorganizowane przez procesor FMC (pod warunkiem, że obszary sekcji są ciągłe), systemowe pakiety stałe SOMU-3 oraz dowolne inne.

Sekcje na takich pakietach traktowane są identycznie jak urządzenia i mogą być przydzielane do strumieni we/wy przy pomocy ekstrakodów ASSIGN i DEFAULT.

7. Przejście z systemu SOM-3 na system DSM-1.

Jedyną poważniejszą trudnością, która może wniknąć przy zastępowaniu systemu SOM-3 przez system DSM-1, jest konieczność przekopiowania informacji zapisanych na pakietach dyskowych pod kontrolę nowego systemu.

Cała operacja upraszcza się znacznie przy możliwości wykorzystania dwóch pamięci dyskowych. Można wtedy potrzebne informacje kopiować bezpośrednio z jednego pakietu wyłennego na drugi.

W początkowym etapie korzystanie z systemu DSM-1 można bezpośrednio wykorzystywać pakiety dyskowe zorganizowane wg. innych zasad niż w systemie DSM-1.

Szczególnie proste jest wykorzystywanie standardowych pakietów SOMU-3 (z sekcjami RD, A10-AD3, AL1-AL4, AM1-AM4) oraz pakietów wg. standardu FMC.

Inne rodzaje pakietów wymagają, przed ich wykorzystaniem, podania definicji znajdujących się na nich sekcji.

W systemie DSM-1 można łatwo odwzorować całą dotychczasową strukturę sekcji dyskowych wykorzystywanych przez system SOM-3, co pozwala na rozpoczęcie pracy pod nowym systemem praktycznie w tej samej sytuacji, co istniejąca dotychczas pod starym systemem.

Po zakończeniu kopiowania i sprawdzeniu działania oprogramowania można już zapisać nowy system DSM-1 na pakiet stały.

Informacje systemowe zajmują jedynie część pakietu, którego reszta wykorzystywana jest najczęściej na sekcje robocze zadań użytkowników.

Można również na pakiecie systemowym zapisywać dowolne informacje, ponieważ jest on zorganizowany identycznie jak wszystkie inne pakiety

wykorzystywane pod kontrolą podsystemu zbiorów.

W systemie DSM-1 można wykorzystać prawie wszystkie procesory systemu SOM-3 i opracowane pod nim programy użytkowe.

Jedynymi różnicami, które mogą nieco utrudnić przeniesienie oprogramowania spod systemu SOM-3 są :

- brak możliwości bezpośredniego przydziału w programach sekcji dyskowych do strumieni we/wy, przy pomocy ekstrakodów DEFAULT i ASSIGN.
W systemie DSM-1 przydziału takiego dokonuje się poprzez przydzielenie sekcji do numeru logicznego przy użyciu ekstrakodu DASSIGN, a następnie przełączenie obiektu przydziału z numeru logicznego na strumień przy użyciu ekstrakodu DEFAULT lub ASSIGN.
Trudność ta można pokonać, tworząc w razie potrzeby dodatkowe strumienie (stałe lub tymczasowe) o nazwach zgodnych z nazwami potrzebnych sekcji, z przydzielonymi standardowo dowolnymi sekcjami dyskowymi.
- Zmieniona interpretacja wartości opcji TX=1; w drugim słowie tablicy UFT.
- Brak ekstrakodów operacji na sygnałach.

Pierwsza różnica jest nieistotna dla programów korzystających z sekcji dyskowych za pośrednictwem strumieni we/wy o już zdefiniowanych przydziałach. Przydziały takie mogą być dokonane przy pomocy procesora JOB przed uruchomieniem programu.

Przejęcie pod system DSM-1 zwiększa znacznie możliwości już opracowanego oprogramowania.

Szczególne znaczenia mają : możliwość automatycznego rozszerzania sekcji przez podsystem zbiorów oraz praca wieloprogramowa.

Ośrodek autorski służy pomocą przy wdrażaniu nowego systemu.

8. Konfiguracja sprzętu.

System operacyjny DSM-1 wymaga co najmniej :

- pamięci operacyjnej 32 k;
- jednej pamięci dyskowej (przy czym istnieje możliwość przejściowego korzystania wyłącznie z pakietu wymiennego);
- jednego urządzenia we/wy na konsolę operatora;
- czytnika taśmy perforowanej.

Rozszerzony zestaw może zawierać :

- większą pamięć operacyjną;

- większa ilość pamięci dyskowych (przy możliwości korzystania z dużej ilości rozróżnialnych wymiennych pakietów dyskowych),
- dowolna ilość drukarek: DZM-180 (lub innych o podobnym działaniu),
- dowolna ilość drukarek: DZM-180-KSR (lub innych urządzeń o podobnym działaniu np. monitorów ekranowych),
- dowolna ilość czytników taśmy perforowanej,
- dowolna ilość dziurkarek: taśmy papierowej,
- inne urządzenia znakowe lub pamięciowe.

Można również korzystać ze specyficznych cech urządzeń np. możliwości przesyłania małych liter, cofania o znak lub korzystać pewne niepożądane cechy urządzenia (np. zamieniać przesyłane małe litery na duże).

Handlery innych urządzeń nie zostały opracowane dla systemu DSM-1, tylko z powodu braku dostępu do tych urządzeń.

W razie potrzeby ośrodek autorski może podjąć się uzupełnienia systemu o dodatkowe handlery.

9. Procesory systemowe.

Pod kontrolą systemu operacyjnego DSM-1 można wykorzystywać większość procesorów systemu SOM-3. Nie daje się jedynie wykorzystać procesory wykonujące specjalne operacje na pakietach dyskowych (przy wartości opcji TX=1, w drugim słowie tablicy UFT).

Ze względu jednak na wady wielu procesorów i ich małe możliwości opracowano szereg nowych procesorów dla systemu DSM-1, z których część jest odpowiednikami procesorów SOM-3, a reszta pozwala w pełni wykorzystać możliwości systemu DSM-1.

9.1. JOBDSM - procesor podstawowy.

Przeznaczony jest do nawiązania komunikacji z użytkownikiem po uruchomieniu zadania oraz do sterowania przetwarzaniem informacji przy pomocy innych procesorów i programów użytkowych.

Procesor umożliwia :

- przygotowywanie warunków do pracy innych procesorów i programów użytkowych.
- Rezerwację i zwalnianie urządzeń.
- Wykonywanie operacji we/wy za pośrednictwem strumieni.
- Kopiowanie standardowych informacji.

- Operacje pomocnicze jak : tworzenie sekcji dyskowych, proste obliczenia na liczbach całkowitych, kontrola aktualnej daty i czasu.

Szczesólnie cenna cecha procesora jest możliwość realizacji makrodirektyw \$DG. Definicje makrodirektyw mogą być umieszczone w lokalnej bibliotece zadania oraz w bibliotece globalnej.

Globalne makrodirektywy mogą być wykorzystywane przez wszystkie zadania.

Biblioteki definicji makrodirektyw mogą być sekwencyjne lub słownikowe (tworzone przez procesor XIKUSH).

Biblioteki słownikowe zapewniają dużą szybkość wyszukiwania definicji makrodirektyw oraz łatwą ich aktualizację.

Standardowy zestaw globalnych makrodirektyw zapewnia przykładowo możliwość wykonywania operacji na lokalnej bibliotece makrodirektyw.

W tekście definicji makrodirektyw można odwoływać się do argumentów podanych przy wywołaniu. Możliwa jest również translacja warunkowa wykorzystująca wartości zmiennych pomocniczych oraz skoki do fragmentów definicji oznaczanych etykietami.

Translacja ta jest podobna do używanej w translatorze MACROASSEMBLER-a. Możliwe jest wywołanie innych makrodirektyw w treści definicji makroinstrukcji jak i również wywołania rekurencyjne.

Zastosowanie makrodirektyw pozwala na prace użytkownikom nie znającym zupełnie systemu operacyjnego ani jego procesorów.

9.2. CHODSM - operacje na pakietach dyskowych.

Procesor przeznaczony jest przede wszystkim do zapisu podstawowych informacji na pakietach dyskowych (inicjacji-pakietów) oraz tworzenia na nich struktury zbiorów.

W szczyślności procesor pozwala wykonać :

- inicjacje pakietów polegającą na zapisie pól adresowych i standardowej informacji na polach danych sektorów.
- Kopiowanie całych pakietów dyskowych.
- Tworzenie i kasowanie zbiorów.
- Testowanie poprawności struktury informacji na pakietach.
- Kopiowanie informacji z dowolnych pakietów na pakiety zorganizowane wg. zasad podsystemu zbiorów systemu DSM-1.
- Bezpośrednie operacje na sektorach i poszczesólnych słowach pakietów dyskowych, umożliwiające odtworzenie informacji zniszczonej wskutek awarii sprzętu.

Powyższe operacje są możliwe do wykonania, pod kontrolą systemu DSM-1,

Jedynie przy pomocy omawianego procesora.

Może być on użyty jedynie w zadaniu poziomym SYSGROUND) o nazwie CHO, co zabezpiecza procesor przed niepożądanym użyciem.

9.3. AKTDSM - aktualizator tekstowy.

Przeznaczony jest przede wszystkim do modyfikacji źródłowych postaci programów i procedur. Może być również wykorzystywany do przetwarzania dowolnych informacji tekstowych, które składają się z wierszy o długości nie przekraczającej 72 znaki.

Najprostszy sposób wykorzystania procesora polega na wczytywaniu poprawianej informacji ze strumienia SI, korektę jej i wyprowadzanie na strumień SO.

Możliwe jest także wykorzystanie tzw. trybu bezpośredniej korekty, w którym po zakończeniu korekty, poprawiona informacja jest ponownie wyprowadzana na strumień SI. Pozwala to w prosty sposób wykonywać wielokrotną korektę.

Adresowanie wierszy poprawianego tekstu może odbywać się poprzez podawanie :

- numeru bezwzględnie,
- numeru względnego w stosunku do ostatnio poprawianego wiersza,
- występującej w wierszu etykiety,
- ciągu znaków występujących w wierszu,
- adresu względnego w stosunku do etykiety lub ciągu znaków.

Przy pomocy procesora można wykonywać następujące operacje na przetwarzanej informacji tekstowej :

- dodawał, usuwał, zamieniał, przestawiał i kopiował wybrane wiersze. Źródłem wierszy poprawek może być przy tym inna informacja tekstowa dostępna za pośrednictwem pomocniczego strumienia.
- Wstawiał ciągi znaków na początku wiersza, przed podanym ciągiem znaków, po podanym ciągu znaków oraz na końcu wiersza.
- Zamieniał ciąg znaków w wierszu na inny.
- Oznaczał wybrane wiersze czteroznakową nazwą identyfikatora.
- Wstawiał numery wierszy do identyfikatorów.
- Usuwał wiersze komentarza.
- Drukował listy odwołań do wybranych nazw.

- Zamieniał nazwy w całym przetwarzanym tekście.
- Zamieniał ciągi znaków w całym przetwarzanym tekście.
- Redagował wiersze (tzn. dostosowywał ilość początkowych spacji do wymagań translatora FORTRAN-u, lub zmniejszał ilość początkowych spacji do jednej).
- Tworzył i aktualizował biblioteki informacji tekstowych.
- Wykonywał operacje na wielu sekcjach w zbiorze jednocześnie.

Bardzo cenną zaletą opisywanego procesora jest możliwość oznaczania testowych wstawek w źródłowych tekstach programów, niezależnie od języka programowania. W razie potrzeby wiersze wstawek mogą być przed translacją zamienione na spacje. Operacja taka nie zmienia numeracji translowanych następnie wierszy.

9.4. MACDSM - translator rozszerzonego języka MACROASSEMBLER.

Translator MACDSM przeznaczony jest do translacji programów napisanych w rozszerzonym języku MACROASSEMBLER.

Zastępuje on całkowicie i spełnia wszystkie funkcje standardowego translatora MACROASSEMBLER-a z systemu SOM-3, działając dwukrotnie szybciej i zajmując mniejszy obszar pamięci operacyjnej (minimum 12 k).

Najważniejsze dodatkowe możliwości translatora MACDSM to :

- możliwość operowania adresami bajtowymi.
- Zwiększona do 8 ilość typów wartości.
- Rozszerzone możliwości translacji warunkowej. Dopuszczalne jest przykładowo operowanie na pojedynczych znakach argumentów makroinstrukcji oraz na wydzielanych automatycznie z argumentów tzw. elementach wyrażeń.
- Możliwość używania w wyrażeniach nawiasów oraz operatorów relacji.
- Dodatkowe operacje na wartościach odpowiadające działaniu rozkazów OR, NR oraz SHC.
- Występowanie operatora podstawienia wartości nazwy lub wyrażenia. Pozwala on na łatwe generowanie nazw i używanie pomocniczych tablic przy translacji.
- Możliwość optymalizacji wielkości programu wynikowego przez automatyczne zamianę rozkazów dłuższych (dwusłowych) na krótkie (jednosłowe).
- Bardzo czytelna sygnalizacja błędów, zawierająca w druku błędnej wiersza i słowny komentarz.

- Możliwość sygnalizacji błędów prawdopodobnych np. braku odwołań do etykiet lub pomyłek typu UJJI, E10-8.

Znaczenie dwóch ostatnich punktów może docenić każdy, kto przy programowaniu w assemblerze wykorzystywał standardowy translator MACROASSEMBLER-a.

9.5. LIBDSM - bibliotekarz sekwencyjny modułów programowych.

Procesor LIBDSM przeznaczony jest do tworzenia i aktualizacji sekwencyjnych bibliotek modułów programowych.

Biblioteka sekwencyjna umieszczona jest na jednej sekcji dyskowej. Moduły programowe są umieszczone na niej jeden za drugim, począwszy od początku sekcji. Koniec biblioteki określony jest przez koniec sekcji (lub znacznik końca zbioru).

Umieszczone w bibliotece moduły programowe identyfikowane są poprzez ich nazwę. W bibliotece nie można umieścić dwóch modułów o tej samej nazwie.

Biblioteka identyfikowana jest poprzez nazwę sekcji dyskowej, na której jest umieszczona.

Przy pomocy procesora LIBDSM można zorganizować biblioteki programów nieskonsolidowanych, podprogramów oraz programów skonsolidowanych (gotowych do wykonania).

Biblioteki takie zajmują najmniej miejsca w pamięci dyskowej, ponieważ nie zawierają żadnych dodatkowych informacji o modułach.

9.6. CATDSM - bibliotekarz słownikowy modułów programowych.

Procesor CATDSM przeznaczony jest do tworzenia i aktualizacji słownikowych bibliotek modułów programowych, w których słownik i moduły umieszczone są na tej samej sekcji dyskowej.

Umieszczone w bibliotece moduły programowe identyfikowane są poprzez 3-znakowe nazwy nadawane im podczas dołączania do biblioteki.

Nazwy te są całkowicie niezależne od rzeczywistych nazw modułów.

W bibliotece można umieścić ten sam moduł wielokrotnie, oznaczając go różnymi nazwami. Można również modułowi już umieszczonemu w bibliotece nadać dodatkowe nazwy.

Ten typ bibliotek nadaje się przede wszystkim do organizowania bibliotek modułów ładowania (skonsolidowanych programów i ich overlay'ów).

W zerowym rekordzie sekcji umieszczony jest słownik biblioteki, a w następujących, aż do końca sekcji, umieszczone są moduły programowe.

Początkowe słowo słownika ma stałą wartość 2 i jest identyfikatorem rodzaju biblioteki. Następne słowo określa numer pierwszego wolnego rekordu sekcji za zapisanymi na niej modułami.

Pozostała część słownika zawiera 127 dwusłownych pozycji.

Każda pozycja dotyczy jednego modułu umieszczonego w bibliotece; pierwsze słowo zawiera 3-znakową nazwę modułu, a drugie numer rekordu sekcji zawierającego początkowy rekord modułu.

Biblioteka identyfikowana jest poprzez nazwę sekcji dyskowej, na której jest umieszczona.

Biblioteki słownikowe tworzone przez procesor CATDSM wykazują pewne zalety w porównaniu do bibliotek sekwencyjnych tworzonych przez procesor LIBDSM.

Dzięki słownikowi dostęp do umieszczonych w bibliotece modułów jest szybki.

9.7. BIBDSM - bibliotekarz słownikowy modułów programowych.

Procesor BIBDSM przeznaczony jest do tworzenia i aktualizacji słownikowych bibliotek modułów programowych; w których słownik i każdy z modułów umieszczony jest na oddzielnej sekcji dyskowej.

Biblioteki tworzone są według zasad procesora LIBDSM i mogą być również aktualizowane przez ten procesor.

Procesor BIBDSM pozwala tworzyć i aktualizować biblioteki programów i podprogramów.

Za programy uważa się moduły, w których żadna z nazw INTERNAL nie jest taka sama jak nazwa modułu.

Za podprogramy uważa się moduły, w których jedna z nazw INTERNAL jest taka sama jak nazwa modułu.

W bibliotece programów moduły identyfikowane są poprzez 6-znakowe nazwy modułów.

W bibliotece podprogramów moduły identyfikowane są poprzez 6-znakowe nazwy INTERNAL modułów. Nazwa INTERNAL zgodna z nazwą modułu uważana jest za nazwę słowną modułu, a pozostałe za dodatkowe.

Moduły innego rodzaju niż rodzaj biblioteki nie można umieścić w bibliotece.

W bibliotece nie można umieścić dwóch modułów o tej samej nazwie, a w bibliotece podprogramów także dwóch modułów mających taką samą nazwę INTERNAL.

Tworzone przez procesor biblioteki nadają się do organizowania bibliotek programów nieskonsolidowanych; podprogramów oraz modułów ładowania (skonsolidowanych programów i ich overlay'ów).

Moduły umieszczone w takich bibliotekach mogą być ładowane do pamięci przy pomocy ekstrakodów OVERLAY i CHAIN.

Każdy z modułów umieszczonych w bibliotece zapisany jest na oddzielnej sekcji. Sekcje mają wielkość rekordu równą 256 słów, rodzaj treści rekordu równy bin i mogą być chronione hasłem podawanym przy umieszczeniu modułu

w bibliotece. Nazwa modułu zapisana jest także jako kod sekcji.

Sekcja zawierająca słownik i sekcje zawierające moduły są umieszczone w tym samym zbiorze dyskowym (mają ten sam zbiór nadrzędny).

Biblioteka jest identyfikowana poprzez nazwę sekcji dyskowej, na której umieszczony jest słownik biblioteki.

Ważną zaletą bibliotek opisywanego typu jest łatwość dodawania, usuwania i zamiany modułów. Aktualizacja nie wymaga zmian w rozmieszczeniu innych modułów w bibliotece.

Również dostęp do modułów jest szybki dzięki słownikowi.

W wielu przypadkach nazwy zawarte w słowniku są wystarczające, aby wyszukać potrzebne moduły.

Ilość modułów, która można umieścić w bibliotece, jest praktycznie ograniczona jedynie obszarem dostępnym na pakiecie dyskowym, ponieważ słownik biblioteki może zawierać do 32766 pozycji.

9.8. BIRDSM - bibliotekarz słownikowy dowolnej informacji.

Procesor BIRDSM przeznaczony jest do tworzenia i aktualizacji słownikowych bibliotek informacji, w których słownik i każda z informacji umieszczone są na oddzielnych sekcjach dyskowych.

Przy pomocy procesora można tworzyć i aktualizować biblioteki dowolnego rodzaju informacji np. : programów i podprogramów, procedur \$D0 procesora JOB, dowolnych informacji binarnych, danych i wyników programów itp.

Umieszczona w bibliotece informacja może składać się ze standardowych rekordów (znakowych i binarnych) o różnej wielkości lub niestandardowych rekordów binarnych o takiej samej wielkości.

Przesyłane przez procesor informacje nie podlegają żadnej interpretacji i jedynym ich ogranicznikiem jest koniec sekcji (lub znacznik końca zbioru i koniec nośnika).

Informacje w bibliotece są identyfikowane poprzez 6-znakowe nazwy.

Ponieważ treść informacji nie jest sprawdzana, można więc te same informacje umieścić w bibliotece wielokrotnie, oznaczając je różnymi nazwami. Można również informacje już umieszczone w bibliotece oznaczyć dodatkowymi nazwami.

Słownik biblioteki umieszczony jest na sekcji o 3-słowych rekordach.

Nazwy w słowniku biblioteki umieszczone są w kolejności alfabetycznej, co pozwala na szybkie ich wyszukiwanie.

Każda z informacji umieszczonych w bibliotece zapisana jest na oddzielnej sekcji dyskowej.

Sekcje mogą mieć różną wielkość oraz rodzaj treści rekordu (z wyjątkiem

rodzajów BIN i COM wymagających rekordów o wielkości 256 słów.
Nazwa informacji zapisana jest dodatkowo jako kod sekcji.

Zapisane na sekcjach informacje mogą być chronione hasłem (przed nieupoważnionym skasowaniem lub zmianą) oraz wartościami bitów słowa opcji sekcji.

Sekcja zawierająca słownik i sekcje zawierające informacje są umieszczone w tym samym zbiorze dyskowym (mają ten sam zbiór nadrzędny).

Biblioteka jest identyfikowana poprzez nazwę sekcji dyskowej, na której umieszczony jest słownik biblioteki.

Ważną zaletą bibliotek opisywanego typu jest łatwość dodawania, usuwania i zmiany informacji. Aktualizacja nie wymaga zmian w rozmieszczeniu innych informacji w bibliotece.

Również dostęp do informacji jest szybki dzięki słownikowi.

W wielu przypadkach nazwy zawarte w słowniku są wystarczające, aby wyszukać potrzebne informacje.

Ilość informacji, którą można umieścić w bibliotece, jest praktycznie ograniczona jedynie wielkością obszaru dostępnego na pakiecie dyskowym, ponieważ słownik biblioteki może zawierać do 32766 pozycji.

9.9. EDIDSM - konsolidator modułów programowych.

Procesor przeznaczony jest do łączenia modułów programowych i tworzenia z nich modułów ładowania.

Może również służyć do wytwarzania modułów binarnych przeznaczonych do bezpośredniego wprowadzania do pamięci maszyny (przy pomocy klawisza BIN na pulpicie technicznym).

Konsolidator akceptuje moduły wytworzone zarówno przez standardowy translator MACROASSEMBLER-a (z SOM-3) jak i przez procesor MACDSM.

W porównaniu do konsolidatora ED1 z systemu SOM-3 opisywany procesor wytwarza moduły o około 30% mniejszej objętości (dzięki optymalizacji postaci programu wynikowego).

Oprócz operacji podobnych do wykonywanych przez standardowy konsolidator ED1, opisywany procesor umożliwia dodatkowo:

- tworzenie overlay'ów (nakładek) uniwersalnych i specjalnych.
Oprócz tworzenia standardowych overlay'ów możliwy jest również podział na overlay'e dopiero na etapie konsolidacji bez konieczności uwzględniania tego faktu w tekstach źródłowych modułów. Przykładowo overlay'ami mogą być wieloargumentowe procedury i funkcje fortranowskie.
- Definiowanie globalnego obszaru COMMON.

- Automatische określanie adresów i wielkości roboczych tablic dla podprogramów na podstawie analizy adresu (tablice takie deklarowane są bezpośrednio STA rozszerzonego MACROASSEMBLER-a procesora MACDSM).
- Wydruki informacji o łączonych modułach.

Konsolidator umożliwia dołączanie modułów z bibliotek sekwencyjnych oraz słownikowych (tworzonych przez procesory BIBDSM i MIRDSM).

Szczególnie korzystne jest wykorzystywanie bibliotek słownikowych pozwalających na szybkie wyszukiwanie brakujących modułów.

Istotną zaletą konsolidatora jest bardzo prosta i czytelna sygnalizacja występowania niezdefiniowanych nazw external oraz możliwość wydruku informacji o zbędnych definicjach INTERNAL.

9.10. FSYDSM - obsługa podsystemu zbiorów dyskowych.

Procesor FSYDSM pozwala wykonywać wszystkie operacje na sekcjach za pośrednictwem ekstraktów systemowych.

W wielu przypadkach operacje wykonywane przez opisywany procesor wystarczają, aby umożliwić tworzenie i aktualizacje prostych baz danych.

Bardzo użyteczną możliwością jest także drukowanie rekordów sekcji (lub tylko ich wybranych fragmentów) w dowolnej definiowanej postaci.

Pozwala to uniknąć konieczności pisania wielu pomocniczych programów sprawdzających postacie zapisów do sekcji dokonywanych przez uruchamiany zestaw oprogramowania.

Przy pomocy procesora można wykonywać :

- tworzenie i kasowanie sekcji w zbiorach.
- Zmianę wymiarów sekcji, jej słowa opcji, czasu ochrony oraz kodu sekcji.
- Rezerwacje i zwalnianie rezerwacji sekcji.
- Zapis, usuwanie, wstawianie oraz przestawianie wskazanych rekordów sekcji.
- Kopiowanie zawartości jednej sekcji dyskowej do drugiej.
- Tworzenie kopii wszystkich sekcji zbioru (lub tylko niektórych) w innym zbiorze.
- Wydruki informacji o zbiorze przydzielonym.
- Wydruki informacji o sekcjach w zbiorze przydzielonym.
- Wydruki całych rekordów sekcji (lub ich części) w różnych postaciach.

10. Orientacyjne ceny sprzedaży systemu i procesorów.

Wersja podstawowa systemu -----	150	tys.
CHODSM - operacje na pakietach dyskowych -----	--	
JOBISM - procesor podstawowy -----	--	
zestaw makrodyrektyw procesora JORDSM -----	--	
AKIISM - aktualizator tekstu -----	30	
MACISM - translator rozszerzonego języka MACROASSEMBLER -----	40	
LIBISM - bibliotekarz sekwencyjny modułów programowych -----	20	
CATISM - bibliotekarz słownikowy modułów programowych -----	20	
BIBISM - bibliotekarz słownikowy modułów programowych -----	20	
BIRISM - bibliotekarz słownikowy dowolnej informacji -----	20	
EDIDSM - konsolidator modułów programowych -----	30	
FSYISM - procesor obsługi zbiorów dyskowych -----	20	
	RAZEM	350, tys.

M. Guja, K. Jójczyk
Katedra Informatyki
Uniwersytet Jasielloński
Kraków, Kopernika 27

SYSTEM OPERACYJNY SOM3.P — WERSJA V.2

1. Wstęp

System operacyjny SOM3.P w wersji V.1 wykorzystywany jest w Laboratorium Systemów Cyfrowych od 5 lat. W wersji V.2, przewidzianej do dystrybucji w roku 1986, starano się usunąć wiele niedosadności sygnalizowanych przez użytkowników systemu.

Ze względu na stale obniżaną cenę pamięci operacyjnej coraz rzadziej można spotkać tak popularną dawniej konfigurację podstawowa wyposażona jedynie w 32K słów pamięci operacyjnej. Według dokumentacji technicznej pamięć można rozszerzyć bez żadnych zmian sprzętowych do 1Mbyte. W związku z tym faktem użytkownicy wymagają:

- (p.1) wielu zadań o dynamicznej wielkości pamięci operacyjnej
- (p.2) możliwości wykorzystania obszaru większego niż 64K słów.

Postulat (p.1) został już częściowo zrealizowany w wersji V.1 przez wygenerowanie systemu operacyjnego o określonej liczbie zadań dysponujących ekstrakodem pobierania pamięci operacyjnej w porcjach po 4K słowa oraz ekstrakodem zwalniania jej. Jednakże,

ze względu na obecność Jedyneho uniwersalnego interpretera komend wielkość zadania nie moła być mniejsza niż 20K słowa, zaś ze względu na długość słowa rozkazowego nie moła być większa niż 64K słowa.

Nada rozwiązania przyjęto w wersji V.1 systemu operacyjnego SOM3.P był również przeniesiony z systemu SOM3 statyczny podział pamięci dyskowej. Każde, nawet uśpione zadanie użytkownika miało na stałe zdefiniowaną pewną liczbę sekcji roboczych niezbędnych do Jego standardowego działania (ASA, ASB, i inne). Tym sposobem przestrzeń pamięci zewnętrznej rozpadała się na małe sekcje robocze o ustalonej na etapie generacji długości. Unieemożliwiło to na przykład prace tylko Jednego zadania ale za to z dużymi sekcjami roboczymi.

Ponadto statyczny podział globalnych sekcji roboczych wymuszał rozwiązania sztuczne, polegające na wielokrotnym nazywaniu tego samego obszaru w pamięci zewnętrznej (na przykład obszar dwóch sąsiednich sekcji AA1 i AA2 nazwano inaczej AB1).

2. Koncepcja systemu operacyjnego SOM3.P w wersji V.2

System SOM3.P w wersji V.2 zarządza dynamiczną liczbą zadań użytkowników. Stosowne komendy zadania komunikacji pozwalają na aktywowanie i deaktywowanie kolejnych zadań. Wszystkie zasoby nowokreowanego zadania, a więc pamięć operacyjna, pamięć zewnętrzna, słowo opcji, słowo komunikacji muszą więc być w jakiś sposób definiowane.

Przyjęto, że dla kreowanego zadania należy zdefiniować

- (d.1) Jego trzyznakowa nazwa
- (d.2) nazwę interpretera komend
- (d.3) minimalna wielkość pamięci operacyjnej zadania
- (d.4) nazwę konsoli lokalnej

(d.5) potencjalna liczba strumieni i sekcji roboczych.

Interpreter nowo utworzonego zadania (d.1), w zasobach którego istnieje już podstawowe strumienie 'CI' oraz 'CO', zgłosi się na konsoli lokalnej (d.4), pozwalając środkami makroprocesora języka opisu zadania zdefiniować lokalne sekcje robocze oraz lokalne strumienie zadania. W trakcie pracy można stosownymi komendami interpretera komend redefiniować lokalne zasoby zadania (sekcje robocze, strumienie).

Zwróćmy uwagę na fakt, że w wersji V.2 interpreter komend (d.2) nie jest już tak silnie związany z zadaniem jak w poprzednich wersjach systemu operacyjnego.

2.1. Pamięć operacyjna

Wielkość początkowa pamięci operacyjnej zadania określona jest na podstawie wielkości przyjętego interpretera komend (d.2) oraz minimum zdefiniowanego dla danego zadania (d.3). Każdorazowe ładowanie programu do pamięci operacyjnej może powodować jej zwiększenie - decyduje o tym program ładujący, a potem program ładowany. Pamięć operacyjna zadania zwracana jest do puli pamięci wolnej przez wykonanie stosownej komendy interpretera.

W celu bardziej efektywnego wykorzystania pamięci operacyjnej wprowadzono możliwość nadawania obszarom w pamięci operacyjnej nazw. Obszar - spójnie adresowany fragment pamięci - można dołączyć do zadania użytkownika od wskazanego adresu począwszy. Istnieje możliwość dołączania jedynie wybranego fragmentu obszaru. Nie ma ograniczeń na liczbę zadań wykorzystujących dany obszar w tym samym czasie.

Konstrukcja obszaru pozwala na wykorzystanie nazwanego fragmentu pamięci operacyjnej jako bufora komunikacyjnego pomiędzy wieloma zadaniami. Pozwala też na uruchamianie programów

wieloaktywnych, a więc takich, w których wielu użytkowników korzysta z Jednego kodu programu mając jedynie różne obszary z Jeno danymi.

Wielkość obszaru ograniczona jest jedynie wielkością dostępnej pamięci operacyjnej. Wynika stąd, że sposobem przedstawionym powyżej można adresować pamięć wirtualna o wielkości ograniczonej jedynie fizycznie. Realizowany jest zatem postulat (p.2) założeń systemu.

2.2. Pamięć zewnętrzna

W systemie wersji V.2 zrealizowana z predefiniowania sekcji dyskowych na etapie generowania systemu operacyjnego. Użytkownik sam prowadzi gospodarkę pamięcią dyskową (w podobny sposób jak w systemie =FHC=) kreując odpowiedniej wielkości sekcje slobalne zarówno na kasecie wymiennej jak i na dysku stałym. Doisy wykreowanych w ten sposób sekcji są przechowywane na dysku. Sekcje slobalne wprowadza się do systemu odpowiednią komendą zadania komunikacji.

Pozostały wolny obszar przeznaczony jest do wykorzystania na lokalne sekcje dyskowe zadań użytkowników. Pamięć zajęta przez sekcje lokalne jest zwracana do puli pamięci zewnętrznej na życzenie użytkownika lub podczas kasowania zadania.

2.3. System plików

W wersji V.2 systemu operacyjnego SOH3.P pozostawiono nadal dwupoziomowy system plików. Tak zwana sekcja plikowa mieści się w standardowym pliku pamięci zewnętrznej. Jej wielkość, w przypadku dysku sztywne, może być definiowana przez użytkownika.

Dopuszczalne jest używanie sekcji plikowej na 8" dyskach

elastycznych i zorganizowanie jej w różnych osłonie znanych standardach. Otwierając takie kartoteki (w dyskietkowych systemach plikowych jako nazwa kartoteki wykorzystywana jest nazwa dyskietki) jako opcjonalny parametr można podać nazwę systemu plikowego: RT/11, CP/M, DOS, FDOS. Dalsze komendy dotyczące danej kartoteki (odczyt pliku, archiwizowanie pliku, kasowanie pliku, informacje o kartotece itp.), mimo że zapisane podobnie syntaktycznie, realizowane są zgodnie z zadeklarowanym standardem.

3. Języki programowania

Podstawowym językiem programowania wykorzystywanym przez użytkowników Laboratorium Systemów Cyfrowych jest niestety nadal język Fortran. Zaimplementowany ostatnio preprocesor tego języka pozwala już jednak na wygodne i bezpieczne kodowanie algorytmów. Preprocesorem tym jest RATFOR, znany doskonale użytkownikom minikomputerów serii PDP-11.

Konkurencją dla niego stanowi translator innej wersji strukturalnego Fortranu przeniesiony z komputera CYBER - IFTRAN.

Jednakże oprogramowanie systemowe coraz częściej kodowane jest nie w asemblerze, a w maszynowo zorientowanym języku wysokiego poziomu - BCPL. Laboratorium dysponuje na razie tylko translatores tego języka do asemblera, ale prace nad jego kompilatorem są dość poważnie zaawansowane.

Niestety ze względu na brak dobrego kompilatora nadal nie są wykorzystywane ani Pascal, ani Logo.

Zrealizowano natomiast interpretery języków LISP i PROLOG oraz translator języka SPITBOL. Produkty te wykorzystywane są w dydaktyce informatyki przez studentów Katedry Informatyki UJ.

4. Podsumowanie

System operacyjny SOM3.P w wersji V.2, którego prototypowa wersja pracuje już w Laboratorium Systemów Cyfrowych jest kolejnym krokiem na drodze przybliżania komputera do użytkownika. Wprowadzone kolejno innowacje są z zadowoleniem przyjmowane przez użytkowników. System w tej wersji znacznie optymalniej wykorzystuje zasoby komputera. Nie znaczy to jednak, że jest to wersja ostateczna.

Kolejnym krokiem będzie usunięcie zadania komunikacji z pełnymi konsekwencjami tego kroku oraz wprowadzenie wewnętrznego systemu plikowego. Decyzje te wiążą się z zupełną zmianą interpretera komend języka opisu zadania. Przewidywana instalacja pamięci zewnętrznej o dużej pojemności (30Mbyków) pozwala na optymistyczne spojrzenie w przyszłość.

Janusz Gucałek
Jacek Klauziński
Ryszard Zenker

TRANSLATOR JEZYKA ALGOL - 1204MD
dla minikomputera MERA-400
w systemie operacyjnym CROOK-4

1. Uwagi ogólne.

Przyjmujemy, że czytelnik zna międzynarodowy język algorytmiczny ALGOL 60 w zakresie ogólnie dostępnej książki [1]. Z tego powodu omawiamy tylko sprawy specyficzne dla konkretnej reprezentacji ALGOLu dla maszyny Mera-400.

ALGOL 1204M jest w sensie gramatycznym podzbiorem ALGOLu-60. Rozumiemy przez to, że jeżeli pominiemy pewne szczególne typograficzne związane z możliwościami urządzeń i komputera, to każdy program napisany w ALGOLu-1204M jest pod względem gramatycznym programem w ALGOLu-60.

Niżej podano pełny wykaz ograniczeń obowiązujących w ALGOLu-1204M.

6. Ograniczenia gramatyczne.

6.1. Każdy parametr formalny musi mieć specyfikację.

6.2. Etykieta musi być nazwą.

5. Ograniczenia i zmiany semantyczne.

5.1. Skok przy nie określonym przełoczeniu powoduje zatrzymanie programu i sygnalizację błędu.

5.2. Wynik potęgowania jest typu $\backslash i \backslash t \backslash e \backslash s \backslash e \backslash r$, jeżeli podstawa jest typu $\backslash i \backslash n \backslash t \backslash e \backslash s \backslash r$, a wykładnik jest liczbą całkowitą; w przeciwnym razie wynik potęgowania jest typu $\backslash r \backslash e \backslash u \backslash l$.

1. Ograniczenia ilościowe.

1.1. Nazwy, których 63 początkowe znaki są takie same uważa się za identyczne.

1.2. Wykaz par granicznych w opisie tablic z mianem $\backslash o \backslash w \backslash n$ może zawierać co najwyżej 10 par granicznych.

1.3. Stopień procedury nie może przekraczać 3.

Istnieją również inne ograniczenia ilościowe spowodowane skończoną wielkością rejestrów maszyny i sposobem pamiętania informacji przez translator. Nie wymieniamy wszystkich tych ograniczeń, ponieważ są one tak dobrane, że ich przekroczenie jest mało prawdopodobne bez uprzedniego wypełnienia całej dostępnej pamięci maszyny informacjami o tłumaczonym programie.

Przekroczenie jakiegos ograniczenia jest sygnalizowane z wyjątkiem przekroczenia przedziału liczb całkowitych przy wykonywaniu działań na takich wielkościach.

Z. Zmiany.

- Z.1. Wszystkie słowa kluczowe języka ALGOL-1204M zawierają znaki podkreślenia znak "\ (np. "VARIABLE", "WRITE", itd.)
- Z.2. Język ALGOL-1204M całkowicie zbudowany jest z dużych liter (możliwa jest wersja translatora z małymi i dużymi literami).
- Z.3. Całkowicie język ALGOL-1204M jest zbudowany ze znaków kodu ISO-7.
- Z.4. Operatory arytmetyczne i logiczne języka zastąpiono odpowiednimi słowami kluczowymi (pkt.3).

Zaproponowane sposoby pisania słów kluczowych języka jest jedna z propozycji. Błąd może autorzy zastosuje słowa kluczowe w następujących formach "BEGIN" (podobnie jak ALcol 1305) lub "BEGIN", a może "BEGIN". Jest to problem do rozwiązania w najbliższym czasie.

2. Eksekucja programu.

W systemie operacyjnym CROOK-4, translator języka ALGOL-1204M jest oddzielnym programem użytkownika. Jest on dostępny dla każdego użytkownika zadeklarowanego w systemie. W wyniku kompilacji programu napisanego w ALGOLu uzyskuje się gotowy moduł binarny do natychmiastowego uruchomienia.

Proces kompilacji realizuje następujące zlecenia programu OSL.

XALB [,(TZ WED)](,(TZ EDN)] [,(WO)]...

(WO) = ?LIS
 ?TRA
 ?TPR

Zlecenie uruchamia translację programu napisanego w języku ALGOL-1204M. Parametr pierwszego zlecenia zawiera tytuł zbioru wejściowego z tekstem źródłowym w tym języku. Parametr drugi określa tytuł zbioru binarnego. Zbiór ten jest tworzony (jeżeli nie istnieje) przez translator, a po zakończeniu translacji jest zastępowany. Taki zbiór binarny posiada typ BBL (Binarny ALcol program).

Parametr trzeci określa opcje przebiegu translacji.

Znaczenie opcji:

?LIS

połączenia: umieszczenia na monitor punktów orientacyjnych.

?TRA

połączenia: uruchamia translacji ze śleden dynamicznym i śleden retroaktywnym.

?TPR

połączenia: uruchamia translacji procedury.

Jeżeli zlecenie uruchamia jest bez parametru drugiego to nastąpi jedynie tzw. test translacji. W tym przypadku nie będzie wprowadzonym kod binarny do zbioru.

Zlecenie to można uruchomić bez parametrów. Wtedy translator zsyła się znakami "E". Można wtedy używać jego standardowych zleceń.

3. Proces translacji.

Uruchomienie programu translator wykonuje w sposób natychmiastowy w. podanych parametrów zlecenia programu OSL. Jeżeli w czasie trwe-

1985.10.10

AL60L-1204M

czenia translator wykrył błędy, to sygnałem zatrzymania jest komunikat SWRRY. W przeciwnym razie sygnał zatrzymania ma postać:

pP rR

na przykład p270 r1192 - gdzie P jest liczbą słów maszynowych przekładu programu, a R jest liczbą komórek roboczych dostępnych dla rezerwacji dynamicznej (np. dla tablic i segmentów programu).

4. Ogólne uwagi o działaniu translatora.

Dla lepszego rozumienia i interpretacji sygnałów drukowanych przez translator AL60L-1204M pożyteczna jest pewna ogólna znajomość sposobu pracy tego translatora. Proces tłumaczenia jest wykonywany w dwóch rozłącznych w czasie etapach. Czynnności wykonywane w każdym z tych etapów opisujemy krótko poniżej.

ETAP 1.

Translator czyta tekst programu ze zbioru dyskowego i zapamiętuje go w bardzo dokładnej do dalszej analizy i wymagającej bardzo małej pamięci. W czasie czytania translator analizuje strukturę blokową programu oraz opisy nazw i ich obszary działania. Wyrażenia warunków "dla", warunki "jeśli" rozpoczynające instrukcje i instrukcje podstawowe są w tym czasie tylko przetwarzane i zapamiętywane, bez analizowania ich poprawności gramatycznej. Opisy zmiennych prostych, wykazy parametrów formalnych, zbiory wartości i specyfikacje są analizowane dokładnie, a następnie usuwane z tekstu programu. Etap 1 kończy się praktycznie z chwilą przeczytania zbioru z dysku. W czasie działania etapu 1 są wykrywane niektóre błędy formalne znajdujące się w programie. Etap 2 działa wtedy, gdy w etapie 1 nie wykryto żadnego błędu.

ETAP 2.

W celu sprawdzenia, czy tekst przedstawiony do tłumaczenia jest programem, translator wykonuje pełny rozbiór gramatyczny zapamiętanego tekstu wykorzystując utworzone w etapie 1 informacje o nazwach i ich obszarach działania. Jednocześnie program jest tłumaczony na kod wewnętrzny maszyny, przy czym przekład częściowo optymalizuje się w celu odpowiedniego wykorzystania możliwości maszyny. W czasie działania etapu 2 może ulec zniszczeniu zapis etapu 1 w pamięci maszyny. Dzieje się to jednak tylko w razie istotnej potrzeby, przy tłumaczeniu większych programów.

Po zakończeniu tłumaczenia następuje przejście do stanu STOP. Sygnał zatrzymania zawiera informacje o objętości przekładu i zapasie wolnych komórek pamięci.

Translator w czasie tłumaczenia programów nie wykorzystuje dysku. Obydwa etapy są organizowane w pamięci maszyny.

5. Eksploatacja programów użytkowych.

Programy binarne, które powstały w wyniku translacji zleceń XALB są ładowane do pamięci operacyjnej i jednocześnie uruchamiane specjalnym loader'em. Dla uruchomienia tego loader'a służy zlecenie języka operatora RALG (Run program AL60L).

RALG [.(TZ BIN)] [.(NO)]...

(NO) = ?TRA
?DIY
?W

?W0

?M1

pierwszy parametr zlecenia określa tytuł zbioru binarnego a pozostałe parametry oznaczają opcje uruchomienia programu.

Jeżeli podany tytuł zbioru binarnego nie będzie zbiorem binarnym algotowskim (utworzonym zleceniem XALG) i nie będzie posiadał typu BAL (Binary ALgol program) to sygnalizowany będzie komunikat LOAD ERROR. Programu takiego nie można załadować do pamięci.

?TRA

polecenie uruchomienie programu ze śladem.

?BIT

polecenie drukowania na monitorze 10 ostatnich wierszy śladu dynamicznego .. ślad będzie drukowany jeżeli wystąpił błąd w programie.

?W

polecenie wykonania instrukcji algotowskich: setinput(I); setoutput(I); i uruchomienie programu od początku.

?W0

polecenie wykonania instrukcji algotowskich setinput(0); setoutput(0); i uruchomienie programu od początku.

?M1

polecenie wykonania instrukcji algotowskich setinput(I); setoutput(I); i uruchomienia programu od początku.

Standardowo użyta jest opcja ?W0. (tzn. czytanie i pisanie z/na monitor).

Uwaga:

Każdorazowe użycie instrukcji setinput(I); powoduje przełączenie czytania na strumień "WE". Analogicznie użycie instrukcji setoutput(I); powoduje przełączenie pisania na strumień "WY".

6. Alfabet Języka AL60L-1204MD.

- Każdy tekst algotowski zapisany jest przy użyciu ściśle określonych symboli, zwanych symbolami podstawowymi.

Zbiór symboli podstawowych Języka wzorcowego jest stosunkowo bogaty - składa się z małych i dużych liter alfabetu angielskiego, cyfr oraz innych znaków kodu ISO-7. Składa się on również z ponad 30 wyrazów angielskich zawierających znak "\ " przed każdą literą w wyrazie (\b\ \n\ \r\ \t\ itd.). Są to tzw. słowa kluczowe języka.

Przedstawiony niżej alfabet języka AL60L-1204MD jest zapisany w notacji języku i znacznie się różni od języka wzorcowego AL60L 60 [1] i języka AL60L-1204 dla maszyny Odra-1204 [2]. W szczególności zawiera uleszys operatory arytmetyczne, relacji i logiczne.

Opis w metajęzyku.

```

{symbol podstawowy} ::= {litera} | {cyfra} |
                        {wartosc logiczna} | {osroanicznik}

{litera} ::= a|b|c|d|e|f|g|h|i|j|k|l|m|n|o|p|q|
             r|s|t|u|v|w|x|y|z|
             A|B|C|D|E|F|G|H|I|J|K|L|M|N|O|P|Q|
             R|S|T|U|V|W|X|Y|Z|

{cyfra} ::= 0|1|2|3|4|5|6|7|8|9

{wartosc logiczna} ::= \t|\u|\e | \f|\a|\s|\e

{osroanicznik} ::= {operator}|{przerwywnik}|{nawias}|{miano}|
                  {specyfikator}

{operator} ::= {operator arytmetyczny}|{operator relacji}|
               {operator logiczny}|{operator nastepstwa}

{operator arytmetyczny} ::= + | - | * | / | \d|\i|\v | **

{operator relacji} ::= \l|\t | \l|\e | = | \n|\e | \s|\e | \s|\t

{operator logiczny} ::= \e|\v|\v | \i|\w|\b | \o|\r | \a|\n|\d |
                       \n|\o|\t

{operator nastepstwa} ::= \s|\o | \t|\o | \i|\f | \t|\h|\e|\n |
                       \e|\l|\s|\e | \f|\o|\r | \d|\o

{przerwywnik} ::= ,|.|!|:|;|}|::|=| | \s|\t|\e|\p | \u|\n|\t|\i|\l |
                 \w|\h|\i|\l|\e | \c|\o|\s|\w|\e|\n|\t

{nawias} ::= (|)|[|]|]|'|" | \b|\e|\s|\i|\n | \e|\n|\d

{miano} ::= \o|\w|\n | \b|\o|\o|\l|\e|\a|\n | \i|\n|\t|\e|\s|\e|\r |
            \r|\e|\a|\l | \a|\r|\r|\a|\s | \s|\w|\i|\t|\c|\h |
            \r|\r|\o|\c|\e|\d|\s|\r|\e

{specyfikator} ::= \s|\t|\r|\i|\n|\s | \l|\a|\b|\e|\l | \v|\e|\l|\u|\e

```

7. Przykład programowania w Algolu-1204MD.

Ponieważ ograniczenia Algolu-1204M nie są zbyt kłopotliwe, więc znane z podręczników Algol-60 przykłady programowania są ważne w ALGOL-1204M. Istnieją jednak pewne różnice wynikające z maszyny i systemu operacyjnego (wspomniane wcześniej).

Programy w tym języku przygotowuje się w podobny sposób jak dowolny tekst, tzn. można je wpisywać do zbioru dyskietek listować poprawiać dostępnymi zleceniami programu OSL.

Każdy tekst w języku ALGOL musi być zakończony znakiem "?" podany za ostatnią instrukcją programu od nowej linii. Za znakiem "?" należy podać jeszcze 10 spacji.

Cały alfabet języka składa się z dużych liter. Notowiały słowa kluczowe nie są podkreślane lecz użyty jest do tego znak "\". np. \b|\e|\s|\i|\n, \r|\e|\a|\l itd. Istnieje możliwość przygotowania takiej wersji translatora, który kompiluje programy zbudowane z dużych i małych liter alfabetu.

Istnieją programy źródłowe napisane w ALGOL-1204 w kodzie OPTIM. Można wczytać takie programy (także źródłowe) używając do tego programu OPTISO. Program ten dokona konwersji z kodu maszyny do pisania

1985.10.10

ALGOL-1204MD

DP11HA na kod ISO-7. Wczytany program nadaje się do natychmiastowej translacji (bez dodatkowych zmian) przez translator XALG.

8. Literatura.

- [1]. Stefan Paszkowski
Język Algol-60.
Wydanie dziewiąte, poprawione .
Warszawa 1978 PWN.
- [2]. Krystyna Jerzykiewicz, Jerzy Szczepkowicz;
"ALGOL-1204"
PWN, Warszawa 1973.
- [3]. Zbigniew Czerniak, Marek Nikodemski, Andrzej Bobcowski;
"System operacyjny CROOK-4 dla
minikomputera MERA-400"
IOPG, Gdańsk 1985.

III BAZY DANYCH

PROCESY PROJEKTOWANIA BAZ DANYCH

Stanisław WRYCZA
Uniwersytet Gdański

1. Wprowadzenie

Jednym z charakterystycznych, istotnych trendów informatyki ostatnich kilkunastu lat był rozwój teorii i praktyki w zakresie technologii baz danych. Rozwój ten oznaczał również doskonalenia architektury systemu zarządzania bazami danych /SZBD/ oraz procesów projektowania baz danych. Wynikało to z reakcji na zbyt techniczne traktowanie kwestii tworzenia baz danych, wychodzenia z założeń i ograniczeń konkretnych SZBD a nie z analizy rzeczywistych potrzeb informacyjnych organizacji gospodarczej. Zaprojektowane w ten sposób bazy danych nie spełniały oczekiwań, czy wręcz rozmijały się z założonym celem systemu. Sprowadziło to zagadnienie konstruowania baz danych od kwestii "czy stosować SZBD?" do zagadnienia "jak technologię baz danych stosować efektywnie?"

Celem niniejszego artykułu jest przedstawienie niektórych, mało w kraju spopularyzowanych, propozycji standaryzacyjnych w dziedzinach architektury współczesnego SZBD oraz procesów projektowania baz danych wychodzącym naprzeciw wspomnianemu problemowi. Po przedstawieniu głównych elementów raportu

ANSI/X3/SPARC zaprezentowano etapy cyklu życia bazy danych. W końcowej części opracowania scharakteryzowano proces projektowania baz danych w ujęciu [LUM-79] i odniesiono go do założeń najbardziej uznanych metodyk projektowania baz danych.

2. Architektura systemu zarządzania bazą danych /SZBD/ wg raportu ANSI/X3/SPARC¹

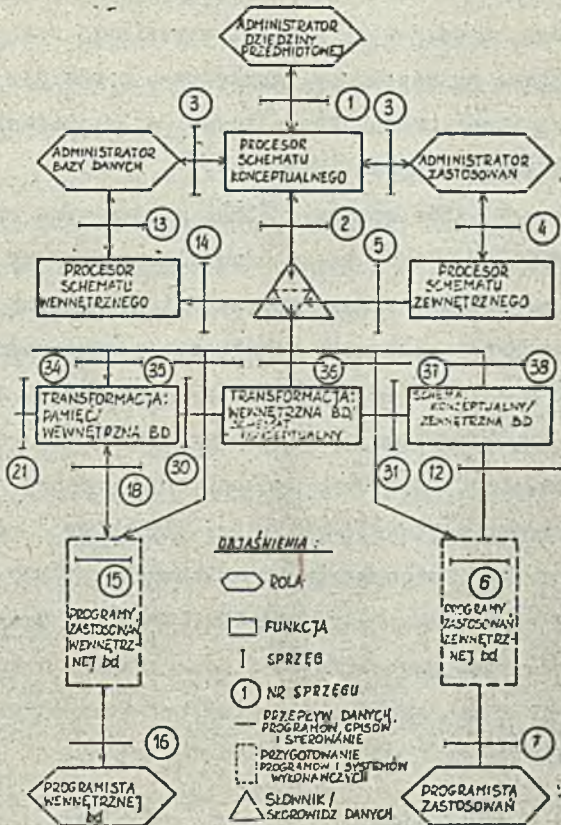
Zaproponowanie przez ANSI architektury SZBD w kolejnych raportach w 1975 [ANS-75] i 1978 r. [TSI-78a] miało zasadniczy wpływ na wyodrębnienie sfery konceptualnego modelowania baz danych. Schemat 1 stanowi koncepcję standardu trójpoziomowego SZBD. Schemat zawiera główne konstrukcje i elementy systemu. Zostały one utrzymane w międzynarodowym standardzie ISO/TC97/SC5/WG3². [GRI-82].

Pełny schemat systemu [TSI-78a, s. 178] obejmuje znacznie bardziej rozbudowany obszar schematu wewnętrznego a więc podział pamięci, wpływ urządzeń komputera na integralność, niezależność i bezpieczeństwo danych, udział podsystemów zapytań, aktualizacji czy generowania zestawień, przesyłania danych na urządzenia pamięci i innych. Kwestie te są konkretyzowane na poziomie wewnętrznym bazy danych, będącym poza zakresem niniejszego opracowania.

¹ ANSI/X3/SPARC - American National Standards Institution Study Group on Database Management Systems.

² ISO/TC97/SC5/WG3 - International Standardizing Organization Working Group on the Aspects of Conceptual Schemata for Data Base Management Systems.

Schemat 1



Struktura SRBD wg propozycji ANSI/X3/SPARC

źródło: [TSI-78a, s. 177].

Głównym celem prac grupy badawczej ANSI/X3/SPARC było określenie zestawu wymagań dla efektywnych SZED. Zasadniczą charakterystyką omawianej propozycji standaryzacji jest oparcie struktury SZED o ściśle wyspecyfikowane i zdefiniowane sprzęgi /ang. interfaces/³, czyli określenie jak poszczególne części systemu są związane, sprzężone, a nie jak części te mają funkcjonować. Pełną listę sprzęgów, w odniesieniu do schematu 1 zawiera praca [TBI-78a].

Trzy poziomy albo schematy SZED, proponowane przez grupę badawczą ANSI/X3/SPARC noszą nazwy: konceptualny /ang. conceptual/, zewnętrzny /ang. external/ i wewnętrzny /ang. internal/. Dwa ostatnie schematy uwzględniają odpowiednio wymagania zastosowań /perspektywa programisty/ bazy danych i systemu komputerowego /perspektywa systemu/ względem SZED, natomiast schemat konceptualny, to sformalizowany opis badanej rzeczywistości, zawierający definicje obiektów, atrybutów i relacji. Wprowadzenie poziomu konceptualnego do struktury SZED stanowi istotną różnicę w stosunku do propozycji CODASYL. Ogólnej charakterystyki schematów zewnętrznych i wewnętrznego dokonujemy poniżej, natomiast schemat konceptualny omówiono oddzielnie w rozdziale II.

a. Schemat konceptualny

Schemat konceptualny w tym ujęciu reprezentuje opis organizacji /ang. enterprise/ równoważny modelowanej bazie danych, przy

³ Chodzi tu raczej o sprzęgi logiczne. Można używać określenia sprzężenia, które jest jednak wykorzystywane w cybernetyce w innym sensie.

czym organizacja oznacza wybraną część świata rzeczywistego. Jej opis obejmuje informacje o zespołach ludzi, rzeczy, idei, zdarzeń i procesów zorganizowanych dla osiągnięcia wspólnych celów. Schemat konceptualny zawiera precyzyjne definicje obiektów, atrybutów i relacji.⁴ Elementy te oraz ograniczenia dotyczące obiektów i relacji nie mogą wystąpić w bazie danych bez uprzedniego zdefiniowania w schemacie. Omawiany schemat /opis organizacji/ jest podstawą odwołań w przypadku różnic pomiędzy programistą a użytkownikiem w zakresie specyfikacji programowych. Schemat konceptualny winien być określony jednoznacznie w maszynowo-czytelnej postaci, w dobrze zdefiniowanym i potencjalnie zstandaryzowanym języku. Funkcję tą aktualnie spełniają języki opisu schematu konceptualnego [WRY-83]. Schematy zewnętrzny i wewnętrzny są zgodne z opisem konceptualnym. Winien on być przekazany do S²ED, aby służyć wszystkim zastosowaniom /funkcjom/ organizacji.

Tak więc schemat konceptualny spełnia następujące role:

- opis informacji związanych z badaną dziedziną przedmiotową,
- stała platforma odniesienia dla schematów zewnętrznego i wewnętrznego,

⁴ Często zakłada się, iż obiekt, atrybut i relacja to pojęcia pierwotne, nie wymagające definiowania. W niniejszym opracowaniu przyjmuje się, iż:

- obiekt to jednoznacznie identyfikowalny składnik badanej rzeczywistości/badanej dziedziny podmiotowej - organizacji/;
- relacja to związek między dwa lub więcej obiektami w badanej dziedzinie podmiotowej
- atrybut stanowi charakterystykę odpowiedniego obiektu bądź relacji.

Należy zaznaczyć iż powyższe pojęcia opisują jedynie statykę schematu a pomijają dynamikę. Statyka może być zdefiniowana przy pomocy podejścia binarnego bądź rachunku predyktów. Kategoriami opisu dynamiki schematu konceptualnego są funkcje, zdarzenia i operacje [WRY-83].

- podstawa do definiowania dodatkowych schematów zewnętrznych lub modyfikowania czy rozszerzenia istniejących, bez wpływu na schemat wewnętrzny,
- podstawa do modyfikacji schematu wewnętrznego, niezależnych od schematu zewnętrznego,
- mechanizm kontroli zawartości i wykorzystania bazy danych.

b. Schemat zewnętrzny

Odpowiada on uproszczonemu modelowi badanej rzeczywistości w odniesieniu do jednego lub większej ilości zastosowań. Cała sfera zewnętrzna /ang. realm/ obejmuje szereg zewnętrznych schematów bazy danych, z których każdy jest zbiorem triad reprezentujących [TSI-78a, s. 184] obiekty, atrybuty i relacje, będące przedmiotem zainteresowania dla poszczególnego zastosowania. Zatem szereg uwag dotyczących schematu konceptualnego należy odnieść do schematu zewnętrznego. Każdy zewnętrzny obraz bazy danych musi być związany z zewnętrznym schematem bazy, opisującym odpowiednie jego obiekty. Schemat zewnętrzny reprezentujący poszczególne zastosowania, służy programistom zastosowań. Dane zewnętrzne nie są zapamiętywane odrębnie lecz generowane na podstawie danych wewnętrznych pod kontrolą schematu konceptualnego. Różne zastosowania i spojrzenia różnych użytkowników na schemat konceptualny nie naruszają współistnienia różnych schematów zewnętrznych [MER-78, s. 160]. Schemat zewnętrzny pełni dwie funkcje: łączenia programów zastosowań z bazą danych oraz opisu zastosowania dla różnych użytkowników.

c. Schemat wewnętrzny

Schemat ten, zwany również schematem fizycznym czy schematem pamięci, stanowi zespół reguł opisujących, jak baza danych jest fizycznie realizowane na urządzeniach pamięci NIJ-78, s. 224. Zakres i struktura schematu wewnętrznego są ograniczone przez sprzęt i dostępny SZBD. Schemat wewnętrzny składa się z tej części danych opisanych w schemacie konceptualnym, które są wymagane przez zastosowania schematów zewnętrznych. Dane są zdefiniowane w schemacie konceptualnym /metadane/ a ich aktualne wartości przechowywane w schemacie wewnętrznym. A zatem występowanie wartości w schemacie wewnętrznym dla każdego elementu schematu konceptualnego nie jest konieczne. Jednocześnie schemat zewnętrzny nie może być zrealizowany, jeśli nie ma odpowiednich wartości w schemacie wewnętrznym, mimo że istnieje jego opis. Sfera wewnętrzna jest zorientowana na najbardziej efektywne wykorzystanie sprzętu informatycznego zgodnie z potrzebami przetwarzania organizacji. Winna ona reprezentować bieżący poziom technologii pamięci, a zatem winna być zdolna do zmian, aby odzwierciedlać aktualny stan technologii komputerowej. Na poziomie wewnętrznym definiowane są ukierunkowane maszynowo struktury danych, ścieżki dostępu, algorytmy wyszukiwania, odsyłacze /ang. pointers/, kontrole redundancji i inne.

Sprawność obsługi i wykorzystanie bazy danych są zależne od poziomu jej administrowania. Znaczenie tej czynności podkreślają raporty CODASYLu / [COD-74], [COD-78] /, wyróżniając odrębną funkcję administratora bazy danych. Opracowanie

ANSI/X3/SPARC [ANS-75] zaleca 3 główne role w administrowaniu bazy danych /por. schemat 1/: administrator dziedziny przedmiotowej /ang. enterprise administrator/. administrator zastosowań /-is/ /ang. application/s/ system administrator/ i administrator bazy danych /ang. database administrator/. Należy zaznaczyć, że każda z ról może być spełniona przez większą liczbę osób. Administrator dziedziny przedmiotowej, w konsultacji z jej pracownikami i analitykami zastosowań identyfikuje ogólne potrzeby w zakresie istniejących, proponowanych bądź potencjalnych zastosowań. Określa więc on:

- wielkość strumienia informacji w organizacji /dziedzinie przedmiotowej/,
- najlepszy sposób zbierania informacji i najlepszą formułę ich prezentacji,
- zasady integralności i bezpieczeństwa, związane z danymi.

Po zrozumieniu przez administratora dziedziny przedmiotowej potrzeb informacyjnych organizacji i dokonaniu ich syntezy poprzez udokumentowanie zastosowań, przepływu i dostępności informacji, przygotowuje on schemat konceptualny /w języku "opisowym"/ jako model informacyjny organizacji i punkt odniesienia dla dalszego rozwoju bazy danych. Administrator organizacji przewiduje dodatkowe zastosowania i zapewnić stabilność między różnymi zastosowaniami. Administrator dziedziny przedmiotowej współpracuje z pozostałymi administratorami przy opracowywaniu schematów zewnętrznego i wewnętrznego. Poprzez sprzęg 1 administrator dziedziny przedmiotowej przeprowadza konwersację w czasie rzeczywistym z procesorem schematu kon-

ceptualnego, specyfikując opis tego schematu i reguły sprawdzenia poprawności danych. Definiowanie schematu konceptualnego może przebiegać wg dwu równoległych procesów:

- określenia potrzeb w zakresie zastosowań,
- określenia składników schematu konceptualnego.

Zadaniem administratora zastosowań jest opracowanie schematów zewnętrznych, definiujących poszczególne zastosowania bazy danych. Każde zastosowanie obejmuje opis danych w postaci odpowiednich struktur danych. Poszczególne dziedziny zastosowań może zajmować się oddzielny administrator zastosowań przygotowujący adekwatny schemat zewnętrzny. Administrator ten wykorzystuje procesor schematu zewnętrznego w trybie konwersacyjnym dla definiowania schematu zewnętrznego, jego kontroli oraz powiązań ze schematem konceptualnym. Każdy schemat zewnętrzny zapewni dostęp do tej części bazy danych, która jest odpowiednią dla danego zastosowania.

Najbardziej dotąd określoną rolą [CAN-74] jest administrator bazy danych. W ujęciu ANSI/X3/SPARC jako główne zadanie polega na zdefiniowaniu schematu wewnętrznego, jest więc nieco odmienne od powszechnego rozumienia tej funkcji. Opisując schemat wewnętrzny administrator określa strategię wykorzystania pamięci przez system zarządzania bazą danych oraz odpowiedni model danych: hierarchiczny, sieciowy czy relacyjny. Administrator dokonuje przekształceń schematu konceptualnego na wewnętrzny, w związku z czym niezbędny staje się bieżący dostęp administratora do schematu konceptualnego. Określa on również precyzyjne, sformalizowane definicje danych wewnętrznych, ich

organizację, przechowywanie i dostęp. Dle osiągnięcia optimum funkcjonowania, zgodnego z bieżącymi priorytetami przekształca on i reorganizuje schemat wewnętrzny. Administrator bazy danych wyznacza ilość przechowywanych kopii schematu wewnętrznego.

Trójkąt w centrum schematu 1 przedstawia słownik/skorowidz danych, który przechowuje wszystkie definicje i reguły związane ze schematami i transformacjami między nimi. Jest to najbardziej podstawowy składnik oswieżonej struktury, stanowiący meta-bazę-danych, przechowywającą dane o bazie danych. Poza wspomnianą minimalnym zakresem definicyjnym, słownik może zawierać: statystykę wykorzystania obiektów, atrybutów i relacji bazy danych, deklaracje dostępu i bezpieczeństwa, deklaracje restartu, odtwarzania, zliczenia i edycji, opis użytkowników oraz nagłówki i opisy odpowiadające wspomnianym możliwościom.

2. Cykl życia bazy danych

Projektowanie baz danych jest jedną z części cyklu życia bazy danych. Dwie główne fazy w jej cyklu życia to projektowanie i eksploatacja. Szczegółowo stadia cyklu z punktu widzenia projektanta i użytkownika przedstawiają się jak na schemacie 2 [FRY-78a, s. 3] .

Tak więc elementy poszczególnych faz cyklu życia systemu z bazą danych to:

A. Faza projektowania

- A.1 sformułowanie i analiza potrzeb
- A.2 projektowanie logicznej bazy danych
- A.3 projektowanie fizycznej bazy danych

B. Faza eksploatacji

B.1 wdrożenia

B.2 eksploatacja i kontrola

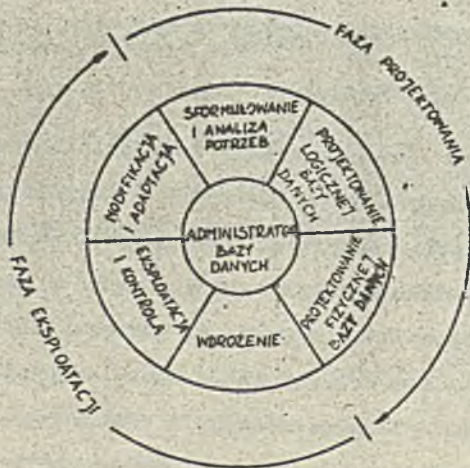
B.3 modyfikacja i adaptacja.

A oto krótka charakterystyka poszczególnych etapów cyklu:

A.1. Sformułowanie i analiza potrzeb

Główne zadanie analizy to zebranie potrzeb informacyjnych od przyszłych użytkowników dla ustalenia zgodności celów systemu z celami użytkownika oraz przepływów informacyjnych w organizacji. Jest to zazwyczaj najbardziej czasochłonny, najtrudniejszy i najważniejszy etap, gdyż wynika z niego większość istotnych decyzji w kolejnych etapach projektowania bazy danych. Jednocześnie jest on najczęściej niewłaściwie realizowanym stadium wywołującym negatywny "kaskadowy" efekt w kolejnych etapach cyklu.

Schemat 2



Cykl życia bazy danych

Źródło: [FRY-78b, s. 1-6]

A.2. Projektowanie logicznej bazy danych

Realizacja tego etapu pozwala na stworzenie bazy danych i związanych z systemem programów. Poprzez konsolidację potrzeb użytkownika następuje zaprojektowanie oraz udoskonalenie struktury informacyjnej. Rezultatem projektowania logicznego jest definicja schematu bazy danych w nomenklaturze raportu CODASYL [COD-71] bądź schematów konceptualnego i wewnętrznego w terminologii ANSI/X3/SPARC [ANS-75] .

A.3. Projektowanie fizycznej bazy danych

Stadium to odpowiadające tworzeniu wewnętrznego schematu w nomenklaturze ANSI/X3/SPARC, zawiera w sobie dwie główne czynności: wybór struktury pamięci oraz przygotowanie do wdrożenia schematu wewnętrznego poprzez specyfikację rozmiarów bloku i przyporządkowanie urządzeń. Wynikiem jest ostatecznie wdrażalny projekt schematu wewnętrznego. Od strony programowej następuje przygotowanie programów w języku manipulacji danymi odpowiednio dla opracowanego projektu logicznej bazy danych.

B.1. Wdrożenie

Etap ten dotyczy stworzenia bazy danych i programów zastosowań, na podstawie projektów logicznej i fizycznej bazy danych oraz wprowadzenia danych do bazy danych. Ta ostatnia czynność jest stosunkowo kosztowna w związku z przekształcaniem ich aktualnej postaci /struktura logiczna i fizyczny format/ na wymaganą przez dobrany SDBD. Zadaniem programów zastosowań jest umożliwienie niezawodnego i efektywnego dostępu do bazy danych dla zaspokojenia potrzeb przetwarzania użytkownika

B.2. Eksploatacja i kontrola

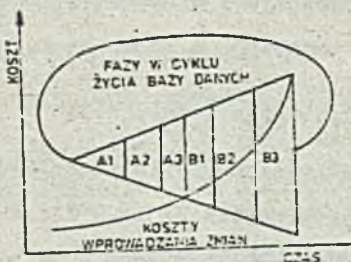
W trakcie eksploatacji systemu następuje ocena jego funkcjonowania w aspekcie aktualności bazy danych wobec potrzeb użytkownika oraz elementów zawodnych w systemie. Integralność bazy danych winna być stale zachowana a błędy bieżąco usuwane.

B.3. Modyfikacja i adaptacja

Celem tego etapu jest udoskonalenie funkcjonowania istniejącego systemu poprzez reorganizację bazy danych i/lub zmianę programów. Zmiany te są rezultatem pojawiania się nowych potrzeb, sygnałów ze stadium eksploatacji i kontroli oraz oceny dotychczasowych rezultatów funkcjonowania systemu przez użytkownika. Reorganizacja bazy danych dotyczy jej fizycznej struktury. Przebiega ona od odwrócenia relacji /restrukturyzacja/do przekształcenia struktury pamięci /zmiana formatów/. Adaptacja programów zapewni otrzymywanie poprawnych wyników mimo reorganizacji bazy danych.

Zarówno czasokres konstruowania bazy danych jak i jej jakość uzależnione są od poprawności i zupełności etapów wstępnych związanych z modelowaniem konceptualnym. Błędy popełnione w tych fazach są zwielokrotnione czasowo i kosztowo w stadium projektowania fizycznej bazy danych /por. schemat 3/.

Schemat 3



Cykl życia bazy danych a koszty wprowadzania zmian

Źródło: Na podstawie [ROS-82, s. 307]

Opisana zależność jest jednym z podstawowych stymulatorów potrzeby rozwoju teorii modelowania konceptualnego.

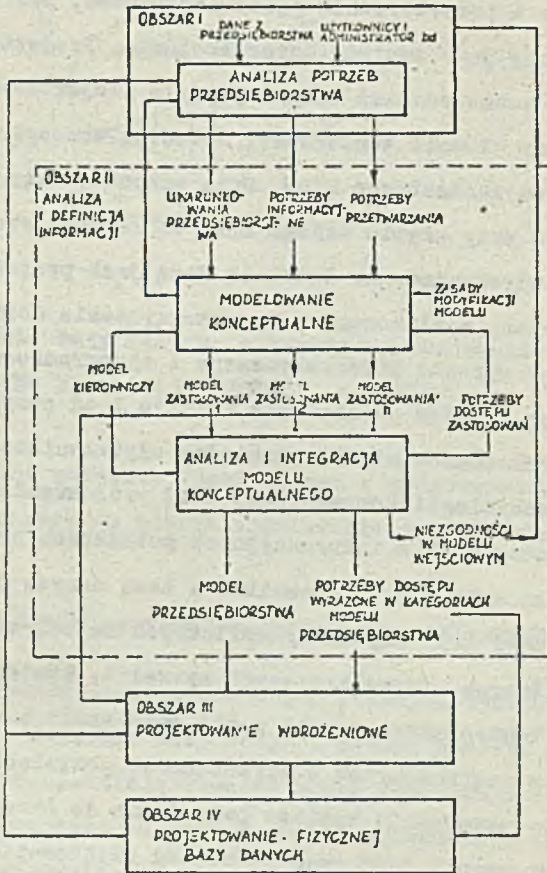
3. Uogólnienie procesu projektowania baz danych

Jak wynika z dotychczasowych rozważań występuje duża różnorodność w zakresie etapizacji i treści procesu projektowania baz danych. Istnieją również pewne podobieństwa między nimi. Wnioski powyższe od pewnego czasu narzucały potrzebę sformułowania ujednoliconych reguł projektowania baz danych wobec wzrostu liczby ich zastosowań w różnych dziedzinach działalności. Najbardziej znaną i uznaną próbą standaryzacji stanowi propozycja grupy 35 autorów, opublikowana w postaci 1978 New Orleans Data Base Design Workshop Report [LUM-79].

Cały proces projektowania przedstawiony na schemacie 4 został podzielony na cztery etapy /obszary projektowe/ choć granice pomiędzy nimi są do pewnego stopnia rozmyte. Obszar I /Analiza Potrzeb Przedsiębiorstwa/ obejmuje określenie potrzeb ogólnych i przetworzenie przedsiębiorstwa, bez uwzględnienia struktury jego systemu informacyjnego. Projektowanie w II obszarze zwane również nieprecyzyjnie projektowaniem logicznym dotyczy głównie konstrukcji schematu konceptualnego, niezależnego od konkretnego SZBD. Opis schematu bazy danych przetwarzanego przy użyciu danego SZBD to zadanie obszaru III - projektowania wdrożeniowego. Ostatnią fazą jest projekt fizycznej bazy danych, realizowany przy wykorzystaniu dostępnego bądź wybranego sprzętu informatycznego i oprogramowania.

Zasadniczym powodem oddzielania obszaru I od obszaru II są trudności w porozumiewaniu się pomiędzy użytkownikami niezorientowanymi w technologii komputerowej oraz projektantami systemów niezorientowanymi w informacyjnych potrzebach przedsiębiorstwa. Komunikowanie się użytkowników z bazą danych dokonuje się w obszarze I przy użyciu specyficznych metod, modeli danych języków czy innych aformalizowanych narzędzi. Stwierdzenia użytkowników są kontrolowane dzięki pętli sprzężenia zwrotnego w obszarze II w postaci pytań o sprzeczności, pominięcia, niejasności oraz przekazywania wyników pośrednich do akceptacji. Główną funkcją obszaru I jest uzyskiwanie od użytkowników informacji o trzech aspektach bazy danych: uwarunkowaniach /ograniczeniach/ przedsiębiorstwa, potrzebach informacyjnych i potrzebach przetwarzania. Pierwszy aspekt stanowi opis funkcjonowania organizacji obejmując czynniki, które ostatecznie mają wpływ na

Schemat 4



Uogólniony proces projektowania bazy danych

Źródło: [LUM-79, s. 337]

projekty logicznej i fizycznej bazy danych oraz dobór sprzętu informatycznego. Potrzeby informacyjne odzwierciedlają sposób ujmowania przez użytkowników struktur danych przedsiębiorstwa i winny obejmować opisy obiektów i relacji między nimi w analizowanej rzeczywistości. Potrzeby przetwarzania obejmują trzy ich rodzaje: planowanie /strategiczny szczebel zarządzania/, kontrolę /szczebel taktyczny/ i operacje /bezpośredni, końcowi użytkownicy systemu/. Faza Analizy Potrzeb Przedsiębiorstwa jest realizowana metodami analizy dokumentów, wywiadów i oceny dotychczasowego wykorzystania maszyn cyfrowych.

Problemy w zakresie wykonania obszaru I można podzielić na cztery kategorie: przygotowanie analityków systemowych, wytwarzanie doskonalszych metodologii i narzędzi analizy systemowej, zorientowanych na użytkownika modeli danych oraz narzędzi i metodologii wspomagających użytkownika w definiowaniu jego własnych potrzeb. Działanie analityków systemowych w mniejszym stopniu opiera się na uporządkowanym, celowym podejściu a bardziej intuicji, poszukiwaniu metodą prób i błędów [WRY-82]. Aktualnie istnieje bądź jest rozwijanych szereg metodyk analizy celów przedsiębiorstwa, jego potrzeb informacyjnych i przetwarzania. Wśród nich należy wymienić w pierwszym rzędzie opracowaną przez grupę badawczą z Uniwersytetu w Sztokholmie na czele z Lundebergiem /[LUN-75], [LUN-78] / metodologię ISAC /Information Systems Work and Analysis of Changes/ oraz metodykę SADT /Structured Analysis and Design Technique/ skonstruowaną pod kierunkiem D.T. Rossa ROS-77, [THO-78] w firmie SofTech. Czynności związane z realizacją obszaru I mogą być

komputerowo-wspomagane np. jak w przypadku stosowania analizatora systemu - pakietu PSL-PSA / [TEI-77] , [TEI-80]/. Kwestie te, nie stanowiące istoty niniejszej pracy, szerzej scharakteryzowane są w cytowanych opracowaniach.

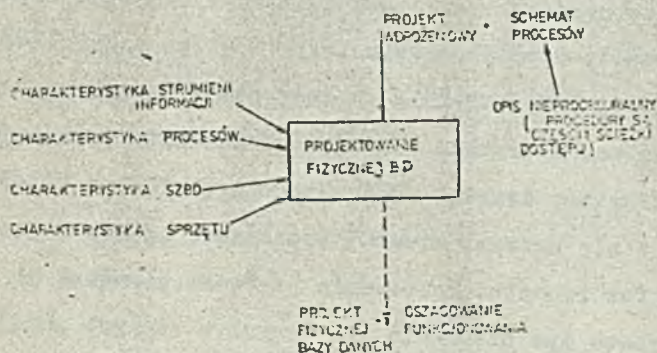
Wyniki prac wykonanych w pierwszym stadium w postaci dokumentów, wypełnionych kwestionariuszy, diagramów i opisów dokonanych w językach sformalizowanych są wprowadzane jako wejście do obszaru II. Główne zadania realizowane w jego obrębie to: analiza i modelowanie informacyjnych potrzeb i ograniczeń przedsiębiorstwa oraz użytkowników w oparciu o odpowiedni model konceptualny dla wytworzenia pożądanych modeli zastosowań i kierowniczych, analiza i modelowanie potrzeb w zakresie przetwarzania oraz integracja modeli zastosowań i kierowniczych we wspólny model przedsiębiorstwa przez wykrycie i usunięcie sprzeczności jak również modyfikacje modeli wejściowych. Rezultatami czynności projektowych wykonanych w obszarze II wprowadzonymi do obszaru III są model konceptualny bazy danych, modyfikowane modele zastosowań niesprzeczne z modelem konceptualnym jak również potrzeby dostępu do bazy danych ujmowane w kategoriach modelu konceptualnego.

Podstawowym zadaniem fazy projektowania wdrożeniowego jest skonstruowanie przetwarzalnego przy użyciu danego SZBD schematu bazy danych, który spełnia wszystkie potrzeby w zakresie użytkowania bazy danych jak: integralność, niezależność, odtwarzalność, bezpieczeństwo i efektywność przy uwzględnieniu wzrostu zakresu i stopnia jej złożoności. Równocześnie z projektowaniem bazy danych następuje specyfikacja założeń i parametrów programów zastosowań korespondujących ze struk-

turą bazy. Rola projektu wdrożeniowego wyraża się w udokumentowaniu w schemacie bazy danych strumieni informacji, potrzeb i uwarunkowań w sposób umożliwiający elastyczność jego przyszłych zastosowań.

Projektowanie fizycznej bazy danych zamyka proces projektowania. Jest to faza najściślej związana z efektywnym funkcjonowaniem i wykorzystaniem bazy danych, toteż najwcześniej stał się on przedmiotem badań, czego rezultatem są m.in. liczne systemy zarządzania bazami danych, Szereg jednak problemów jest w dalszym ciągu przedmiotem intensywnych badań. Elementy wchodzące w zakres projektowania fizycznej bazy danych przedstawione są na schemacie 5. Wynikiem tych czynności jest projekt fizycznej bazy danych akceptowalny przez dany SZBD. Ocena funkcjonowania bazy danych jest produktem ubocznym projektowania w tym obszarze. Stopień szczegółowości projektu

Schemat 5



Projektowanie fizycznej bazy danych

Źródło: [LUM-79, s. 338]

zależy od wymagań SZBD. Reguły i zasady projektowania fizycznego leżą poza zakresem niniejszego opracowania. Bardziej szczegółowe informacje zawarte są m.in. w pracach Cardenasa [CAR-70], Date'a [DAT-81], czy Martina [MAR-83].

Zaprezentowany wyżej standard skłania do porównania istniejących najbardziej uznanych metodyk projektowania baz danych z wyspecyfikowanymi na schemacie 4 obszarami projektowania. W analizie tej wzięto pod uwagę następujące metodyki:

- CADIS-4/[BUB-74], [BUB-75] /
- DSRG / [FRY-78a], [FRY-78b] /
- IBM / [DAT-77], [IMS-77] /
- DBD-DSS / [Gam-77] [GER-78] /
- podejściu infologiczne [SUN-75]
- metodyka Tsichritzisa i Lochovsky'ego [TSI-78b]
- metodyka Kahna [KAH-78].

Jak wynika ze schematu 6 większość metodyk obejmuje wyszczególnione w raporcie nowoorleańskim fazy. W kilku przypadkach /metodyki CADIS-4, IBM, BDB-DSS/ procesy projektowane są wspomagane komputerowo. W trakcie konstrukcji schematu 7 kierowano się istotą i zakresem działań w poszczególnych fazach kolejnych metodyk, a nie nazwami etapów, które choć często różne, są synonimami faz raportu. Porównanie powyższe prowadzi do wniosku, iż proces projektowania baz danych w różnych podejściach jest stosunkowo stabilny i ukształtowany. Znalazł on swoją syntezę w raporcie [LUM-78].

Schemat 6

Lp.	Faza Metodyka	Aplikacja potrzeb	Modelowa- nie kon- ceptualne	Projekto- wanie wirc- tanowe	Projekto- wanie fi- zyczne/bd
1	Metodyka CADIS-4				
2	Metodyka DSRG				
3	Metodyka IBM				
4	Metodyka DBD-DGS				
5	Podejście infolo- giczne				
6	Metodyka Teichrit- sisa i lo- chovaky etc				
7	Metodyka Kotna				



realizacja fazy,

komputerowo wspomaganą realizację

Fazy i metodyki projektowania bazy danych

Źródło: Opracowanie własne

BIBLIOGRAFIA

- [ANS-75] ANSI/X3/SPARC Interim Report, Washington 1975
- [BUB-74] Bubenko J.A. jr., Berild S., CADIS System 4:
A Tool for Incremental Description and Analysis
of Systems, Report TRITA-IBADB, Research Group
CADIS, Stockholm 1974
- [BUB-75] Bubenko J.A. jr., Berild S., Lindencrons - Ohlin
E., Nachmens S., Information Analysis and Design
of Data Base Schema, Research Group CADIS, Stock-
holm 1975
- [GAN-74] Canning R.G. /ed./, Problem Areas in Data Manage-
ment EDP Analyzer, March 1974, vol. 12, nr 3, ss.1-13
- [GAR-79] Cardenas A., Data Base Management Systems, Allyn
and Bacon, Boston 1979
- [COD-71] CODASYL Systems Committee, Report on the CODASYL
Data Base Task Group, April 1971
- [COD-78] CODASYL Data Base Specifications, 1978
- [DAT-77] Data Base Design Aid, Version 2, Designers Guide,
CH20 - 1627 - 2, IBM Technical Publications, White
Plains, 1977
- [DAT-78] Database Technology Infotech State of the Art
Report, Maidnhead 1078, vol. 1: Analysis and
Bibliography; vol. 2: Invited Papers
- [DAT-81] Date C.J., Wprowadzenie do baz danych, WNT, War-
szawa 1981

- [FRY-78a] Fry J.P., Teorey T.J., Design and Performance Tools for Improving Database Usability and Responsivness- Technical Report 78DB 10, Database Systems Research Group, Gradaute School of Business Administration, The University of Michigan, Ann Arbor 1978
- [FRY-78b] Fry J.P., Teorey T.J., Oberlander L.B., Database Design Tools and Metods: A Survey and Tutorial, Technical Report 78DB 14, Database Systems Research Gropu, Gradaute School of Business Administration, The University of Michigan, Ann. Arbor, October 1978
- [GAM-77] Gambino T.J., Gerritsen R., A Database Decision Support System, Department of Decision Sciences, The Wharton School, University of Pennsylvania, Working Paper 77-04-02, 1977
- [GER-78] Gerritsen R., Tools for the Automation of Database Design, The Wharton School, University of Pennsylv-
ania, Working Paper 78-04-02, 1978
- [GRI-82] Griethuysen J.J. i inni /eds/, Concepts and Termi-
nology for the Conceptual Scheme and Information
Base, ISO TC97/SC5/WG3, February 1982
- [IMS-77] IMS/VS Verision 1. General Information Manual,
GM20-1260-5, IBM Corporation, Programming Publis-
hing, San Jose, April 1977
- [KAH-78] Kahn B.K., A Structured Logical Database Design
Methodology, Proceedings New York Data Base Design
Symposium, New York 1978

- [LUM-79] Lum V.Y. i inni, 1978 New Orleans Data Base Design Workshop Report, w: [PRO-79, ss. 328-339]
- [MAR-83] Martin J., Organizacja baz danych, PWN, Warszawa 1983
- [MER-78] Merz L.I., How to Increase Data Base Programing Productivity, w: [DAT-78, ss. 151-183]
- [NIJ-78] Nijssen G.M., The Next Five Years in Data Base Technology, w: [DAT-78, ss. 213-256]
- [PRO-79] Proceedings Very Large Data Bases, Fifth International Conference on Very Large Data Bases, Rio de Janeiro, October 3-5, 1979
- [STR-78] Structured Analysis and Design, Infotech State of the Art Report, Maidenhead 1978, vol. 1: Analysis and Bibliography, vol. 2: Invited Papers
- [ROS-82] Rosenquist C.J., Entity Life Cycle Models and their Applicability to Information Systems Development Life Cycles. A Framework to Information Systems Design and Implementation, The Computer Journal, 1982, vol. 23, nr 2, ss. 307-316
- [ROS-77] Ross D.T., Structured Analysis /SA/: A Language for Communicating Ideas, IEEE Transactions on Software Engineering, vol. SE-3, January 1977, ss. 16-34
- [SUN-75] Sundrgen B., Theory of Data Bases, Petrocelli Charter, New York 1975
- [THO-78] Thomas M., Functional Decomposition: SADT, w: [STR-78, ss. 335-354]

- [TSI-78a] Tsichritzis D., Klug A., ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems, Information Systems, 1978, vol. 3, ss. 173-191
- [TSI-78b] Tsichritzis D., Lochovsky F.H., Designing the Data Base, Datamation, August 1978, ss. 147-151
- [WOJ-83] Wojdyła J., Sobieska-Karpińska J. /red./, Wspomaganie komputerem tworzenia systemów informatycznych /materiały z konferencji/ AE Wrocław 1983
- [WRY-82] Wrycza S., Problemy wyboru i stosowania metodyki analizy wstępnej w nauczaniu projektowania systemów informatycznych, ZN Instytut CEI, UG 1982, nr 5
- [WRY-83] Wrycza S., Języki opisu schematu konceptualnego, w: [WOJ-83, ss. 53-61]

mgr inż. Jan Wierzbicki
Huta Szkła Okiennego
"Szczakowe"
w Jaworznie

PRZEGLĄD SYSTEMÓW OBSŁUGI I ORGANIZACJI
ZBIORÓW DANYCH NA MERA-400

W ostatnich latach notuje się gwałtowny wzrost liczby komputerów jak również dziedzin w jakich znajdują zastosowanie. Podstawową dziedziną technologii komputerowej, umożliwiającą nowe zastosowania, w większości o fundamentalnym znaczeniu jest przetwarzanie danych.

Informatyczne systemy przetwarzania danych są eksploatowane nie tylko w przedsiębiorstwach lecz coraz częściej w różnego rodzaju instytucjach, biurach projektowych, uczelniach itp.. Ułatwiają one w dużym stopniu prace ewidencyjne, księgowo, rozliczeniowe, informacyjne i inne.

Tworzenie takich systemów wymaga od autorów zastosowania określonej organizacji składowania danych w maszynie cyfrowej.

Ogólnie rzecz biorąc istnieją dwa rodzaje działań w tym kierunku. W jednych przedsiębiorstwach najczęściej w początkowej fazie eksploatacji komputera przyjmuje się koncepcję działań ustalającą organizację zbiorów danych lub wręcz decyduje się na jeden z systemów obsługi i organizacji zbiorów.

W innych przedsiębiorstwach grupy projektantów i programistów opracowują niezależne pakiety programów użytkowych, gdzie w rezultacie tworzy się zbiór autonomicznych systemów użytkowych. Nie posiadają one wspólnej płaszczyzny organizacji danych, a poza tym istnieje niebezpieczeństwo ich upadku w przypadku braku autorów. Praktycznie niemożliwa jest wymiana danych między nimi jak również selektywne zbieranie danych np. do systemów informowania kierownictwa.

Dlatego przedsiębiorstwo powinno zdecydować się na przechowywanie danych operacyjnych w oparciu o zintegrowane bazy danych? Jest wiele odpowiedzi na to pytanie. Ogólna odpowiedź brzmi: zapewnia to przedsiębiorstwu centralne sterowanie danymi opera-

cyjnymi co jest najwęższą zaletą.

Kontrastuje to wyrażnie z sytuacją, w jakiej obecnie znajduje się większość przedsiębiorstw. Mają one zbiory danych często przechowywane na prywatnych taśmach i pakietach dyskowych. Dane operacyjne są więc bardzo rozproszone i najczęściej w ogóle nie próbuje się nimi systematycznie sterować. Prowadzi to często do mańnotrawstwa obszaru pamięci.

Oto niektóre korzyści płynące z faktu posiadania systemu sterowania danymi:

- zmniejszenie redundancji pamiętanych danych,
- uniknięcie do pewnego stopnia problemów niezgodności pamiętanych danych,
- dzielenie zapamiętanych danych pomiędzy wielu użytkowników,
- wprowadzenie ograniczenia dostępu do danych,
- zachowanie integralności danych,
- równoważenie sprzecznych wymagań,
- zapewnienie niezależności danych.

Na MERA-400 powstało kilka systemów organizacji i obsługi zbiorów danych zwanych popularnie, choć niezbyt poprawnie bazami danych. Niektóre dopiero powstają inne są modyfikowane. Ilość i zakres prowadzonych prac świadczy o randze problemu.

Na wniosek Rady Porozumienia Użytkowników MERA-400 w dniach 85.08.29 - 85.08.30 w Zakładowym Ośrodku Informatyki Huty Szkła Okiennego "Szczakowa" w Jaworznie odbyło się sygnalizowane od dawna spotkanie twórców baz danych.

W spotkaniu uczestniczyli:

- Maria i Jerzy Kapcia - Politechnika Gdańska - Baza Selko,
- Andrzej Ziólkowski - Polska Akademia Nauk Instytut Badań Systemowych - Baza BD-83 Vitrin,
- w zastępstwie Bronisława Maciuka pan Koczurkiewicz - Politechnika Śląska - Baza BDMB,
- organizatorzy spotkania Jan Wierzbicki i Maria Perek,
- z przyczyn obiektywnych nie mógł wziąć udziału w spotkaniu Witold Rekuć z Politechniki Wrocławskiej - Baza STEP-400.

Celem spotkania było wzajemne zapoznanie się z aktualnym stanem baz danych, wspólna ich ocena oraz ewentualne wypracowanie wniosków co do dalszych prac nad rozwojem dotychczasowych baz

lub stworzeniem nowej bazy.

Z przedstawionych baz największą popularnością cieszą się bazy SELKO pracująca pod systemem operacyjnym CROOK-4 oraz baza BD-83 Vitrin pod SOM'em-3 i CROOK'iem-4.

1. SELKO

Ostatnie działania w kierunku rozwoju bazy SELKO miały na celu przyspieszenie przetwarzania oraz dostępu do zbiorów, które to cechy nie były najmocniejszą stroną tej bazy. Nie będę się rozwodził nad jej szczegółowym opisem, gdyż można go znaleźć w materiałach pokonferencyjnych, przypomnę tylko najistotniejsze elementy.

Aktualnie istnieje możliwość prostego tworzenia wydruków ze złożonymi obliczeniami na podstawie danych ze zbiorów, z nagłówkami itp. ,przy pomocy dyrektyw bazy, co nie wymaga znajomości żadnego z języków programowania. Jednak takie rozwiązanie powoduje, że przetwarzanie jest stosunkowo wolne.

Drugą możliwością jest tworzenie wydruków przy pomocy specjalnego języka wyższego rzędu RAL, który znacznie przyspiesza przetwarzanie oraz dostęp do zbiorów. W perspektywie myśli się nad możliwością automatycznego przejścia z zadania sformułowanego w języku dyrektyw na zadanie w RAL-u, co spowoduje połączenie cech prostoty tworzenia wydawnictw z szybkością działania. W ramach rozwoju, myśli się również o rozszerzeniu metod dostępu do zbiorów oraz stworzeniu automatycznego zestawiania danych z wielu zbiorów.

2. BD-83 / Vitrin /

Baza BD-83 Vitrin została opracowana z myślą maksymalnego wykorzystania systemu operacyjnego SOM-3, między innymi przez wielodostępny program wprowadzania, poprawiania i wyświetlania danych. Ze względu na organizację zbiorów zapewniającą bardzo szybki dostęp do rekordu / co zgodnie potwierdzili uczestnicy spotkania / oprogramowanie może być stosowane do tworzenia złożonych systemów przetwarzania danych.

W tym roku baza danych BD-83 została przerobiona na system operacyjny CROOK-4. Baza ta działa również pod mutantami fabrycznymi SOM'a, pod systemem SOM3 pałacowym oraz po wprowadzeniu drobnych poprawek pod systemem DSM-1. Działa poprawnie w konfi-

guracjach Mera 400 posiadających jednostki grupowe Mera 7900. W bazie pod CROOK-iem w porównaniu z wersją somowską zrezygnowano z wielodostępu w programie obsługi danych gdyż załatwia to system operacyjny. Wprowadzono również kontrolę zakresów wprowadzanych danych.

Bardzo istotną cechą dla użytkowników pragnących przejść na CROOK-a jest fakt identyczności zbiorów tzn. zbiory można przegrywać jeden do jeden z SOM-a na CROOK-a i odwrotnie.

Dzięki temu można przejść z jednego systemu na drugi w każdej chwili /oczywiście pod warunkiem dostosowania fortranowskich programów użytkowych/.

Nie będę przedstawiał charakterystycznych cech tej bazy jednak trzeba przypomnieć, że oprócz funkcji obsługi danych, kopiowania i poprawiania struktur zbiorów, drukowania w odpowiednich układach zestawień ze zbiorów oraz łączenia zbiorów, które to funkcje realizuje 5 standardowych programów BD-83, jest możliwość przy pomocy standardowych dyrektyw /tzw. procedur bazy danych/ tworzyć użytkowe programy /pisane w języku FORTRAN/ które mogą korzystać z danych zawartych w max. 8 zbiorach. Dzięki temu istnieje bardzo elastyczny mechanizm tworzenia dużych i skomplikowanych systemów użytkowych.

BD-83 jest stosowana w przedsiębiorstwach od 1980 roku i jej ciągły rozwój pod kątem potrzeb użytkowników sprawił, że obecnie stała się pewnym i wygodnym narzędziem do tworzenia i obsługi systemów przetwarzania danych.

Podsumowując obie bazy należy powiedzieć, że każda z nich ma swoje zalety i wady.

SELKO - z jednej strony łatwość tworzenia wydruków bez znajomości żadnego języka programowania, z drugiej strony mniejsza efektywność działania powyżej 1000 - 2000 dokumentów w zbiorze. BD-83 - z jednej strony konieczność pisania programów użytkowych w FORTRANIE dla bardziej skomplikowanych wydruków, z drugiej stosunkowo duża szybkość działania na zbiorach do 10 tys. dokumentów /przy max. 32 tys./ i możliwość pracy pod kilkoma systemami operacyjnymi przy zachowaniu identyczności zbiorów.

3. BDMB

Kolejna z baz BDMB opracowana przez mgr Bronisława Maciuka na Politechnice Śląskiej jest wielodostępnym programem zakładowania i aktualizacji kartotekowych zbiorów danych.

Obecnie jest wdrażana w Przedsiębiorstwie Przerobu Żłomu w Gliwicach wraz z systemem rozliczeń dostaw żłomu.

Baza ta choć nie cieszy się jeszcze popularnością, jednak ze względu na niektóre ciekawe rozwiązania może zwrócić na siebie uwagę szczególnie użytkowników SOM-a.

Program BDMB w trybie wielodostępnym dopuszcza równoczesną pracę 4 użytkowników bezpośrednich, którzy mogą korzystać ze wszystkich sposobów przetwarzania danych tj. wyszukiwania, aktualizowania i usuwania danych, sortowania plików, drukowania oraz wywoływania programów użytkowych. Pola dokumentów mogą być grupowane w oddzielnych rekordach, które pamiętane są w oddzielnych plikach, co umożliwia tworzenie kartotek wieloindeksowych.

Przy sortowaniu plików zastosowano metodę wielokierunkowego łączenia wyważonego w połączeniu z sortowaniem w pamięci operacyjnej metodą Hoare-a. Trudno jednak ocenić tę bazę z punktu widzenia użytkownika ze względu na zbyt małą liczbę wdrożeń, jednak już teraz można powiedzieć, że wieloindeksowość, słowniki kodów i prawdopodobnie szybkość działania będą zaletami, a fakt pisania programów użytkowych w assemblerze i języku pre-dykatów - wadą.

4. STEP-400

Kolejną bazą, którą chciałem przedstawić jest System Zarządzania Bazą Danych STEP-400 opracowany przez dr inż. Witolda Rekucia z Politechniki Wrocławskiej.

Jest to pakiet programów i procedur służących do definiowania struktury bazy danych, przygotowywania programów użytkowych oraz kontroli i reorganizacji bazy danych. STEP-400 działa pod systemem operacyjnym SOM-3, a bazowym językiem programowania jest SYMBOL-400. Struktura zbiorów ma charakter sieciowy. Jest ona typowa dla systemów baz danych. Podstawową jednostką danych jest rekord będący ciągiem zmiennych prostych i tablic jednowymiarowych.

Każdy rekord zapamiętany w bazie danych posiada identyfikator wewnętrzny będący unikalnym adresem tego rekordu.

Na określonych typach rekordów /zbiorach rekordów jednakowego typu/ można zdefiniować relację zwaną typem SET-u.

SET jest strukturą charakterystyczną dla podejścia sieciowego, w którym jednemu rekordowi danego typu przyporządkowane jest n rekordów innego typu. W danym typie SET-u jeden typ rekordu pełni rolę właściciela typu SET-u /OWNER/, drugi typ rekordu - rolę członka danego typu SET-u /MEMBER/.

Powiązania rekordów w SET-ach realizowane są przy pomocy łączy adresowych. Rekordy poszczególnych typów pamiętane są w jednostkach pamięci bazy danych zwanych obszarami. Obszar podzielony jest na fragmenty zwane segmentami, w których zapisywane są rekordy tego samego typu.

Strukturę bazy danych definiuje się przy pomocy języka opisu danych JOD. Do specyfikacji operacji w bazie danych służy język manipulacji danymi JMD, którego komendy wstawiane są do tekstu procedur pisanych w języku SYMBOL-400 na zasadach obowiązujących dla instrukcji tego języka.

STEP-400 nie posiada jednak uniwersalnych programów wprowadzania danych jak również drukowania zawartości zbiorów danych.

Programy takie należy pisać dla każdego systemu przetwarzania oddzielnie, a więc mają one charakter użytkowy.

Uniwersalny System Zarządzania Bazą Danych jest więc narzędziem rozszerzającym język programowania SYMBOL-400 o możliwości działania na zbiorach.

5. ADA-1

Uniwersalny Pakiet Programów Prowadzenia Kartotek Dyskowych ADA /od Administrowanie DANymi/ został opracowany w pracowni projektowej EGD CNPTKiP /obecna Fabryka Mierników i Komputerów/. ADA-1 była pierwszą wersją pakietu. Zakładano, że mają powstać następne znacznie poszerzające jej możliwości. Z uwagi na to, że nie był on jeszcze prezentowany, trochę szerzej będą przedstawione mechanizmy jego działania.

Podstawą organizacyjną pakietu jest oparcie zawartych w nim programów na jednolitych metrykach opisów zbiorów i danych oraz indeksowym charakterze zbiorów dyskowych.

Pakiet może być eksploatowany pod systemem operacyjnym SOM-3. W skład ADA-1 wchodzi zestaw programów, które ze względu na swe zastosowanie podzielić można na:

- wspomagające projektowanie systemu użytkowego,
- umożliwiające aplikacje systemu użytkowego,
- umożliwiające bieżącą eksploatację systemu,
- pomocnicze.

Użytkowe sytemy oparte na pakiecie ADA-1 mogą poza programami wchodzącymi w skład pakietu zawierać programy napisane przez użytkownika w języku Macroassembler lub Fortran.

Programy pakietu ADA-1 wykonują następujące funkcje:

- planowanie i przydział dostępnych zasobów pamięci dyskowej pod definiowane w systemie zbiory,
- przyrządowanie zbiorom odpowiednich opisów postaci logicznej rekordów,
- zakładanie zbiorów dyskowych typu indeksowego zgodnie z przypisaną im strukturą logiczną,
- aktualizacja dowolnej pozycji rekordu,
- dopisanie, usunięcie rekordu ze zbioru,
- drukowanie dowolnie definiowanych zestawień zbiorczych,
- tworzenie roboczych tablic dostępu na podstawie definicji klucza,
- realizowanie wydruków o charakterze ewidencyjno-kontrolnym.

Zbiory w organizowanych systemach użytkowych zakładane są na sekcjach. Na sekcji możliwe jest założenie do 26 zbiorów logicznych. Każdy z nich identyfikowany jest literą alfabetu z zakresu od A do Z. Przyjęto, że w rekordzie mogą być maksymalnie 23 dane a długość fizyczna rekordu wynosi 64 słowa.

W przypadku systemu użytkowego, w którym długość rekordu logicznego winna być większa niż 64 słowa lub ilość informacji większa niż 23 należy stosować tzw. przedłużenie rekordu, które fizycznie znajduje się w innym miejscu pamięci dyskowej.

Przedłużeniom przypisane są identyczne metryki co częściom podstawowym. W częściach podstawowych natomiast wstawiany jest łącznik opisów - adres przedłużenia struktury.

Procedura planująca podział dostępnych zasobów pamięci dyskowej określa obszar, w którym lokowane mogą być rekordy zakładanych zbiorów. Z uwagi na efektywność wykorzystania tego obszaru, jest

on spójny dla wszystkich typów rekordów. Podstawowym wymaganiem jest jednak, aby rekrdy wszystkich typów były jednakowej długości. Warunek ten jest zrealizowany dzięki aparatowi przedłużeń. Zapewnia on, że wszystkie fizyczne części rekordów są jednakowe. Jedną z podstawowych zasad administrowania obszarem lokacji jest zasada usuwania rekordów. Usunięcie rekordu powoduje wstawienie jego adresu na początek łańcucha wolnych miejsc dzięki temu odpada potrzeba realizacji takich funkcji jak fizyczne usuwanie rekordów czy zacieśnianie obszarów. Tak więc wpisywany ostatnio rekord, fizycznie nie musi znajdować się na końcu obszaru. ADA-1 zakłada jak już wspomniano, że długość rekordu bądź jego części przedłużającej wynosi 64 słowa.

Tak przyjęta zasada dopuszcza, że w przypadku gdy rekord logiczny jest mniejszy od 64 słów to pozostałe słowa są niewykorzystane. Dlatego też w jednym cylindrze może mieścić się max. 96 rekordów logicznych i ich przedłużeń.

Wydaje się, że tak przyjęta zasada jest zupełnie nieefektywna. Inną niewątpliwie ujemną stroną zastosowanego rozwiązania jest pełne przemieszanie rekordów.

System ma możliwość utworzenia tylko jednej tablicy inwersyjnej. Inną istotną wadą jest stała postać indeksu identyfikacyjnego rekordu. Jest nim dana w CAN-kodzie na 3 słowach maszynowych. Z kołei maksymalna długość klucza w tablicy inwersyjnej może wynosić max. 7 słów maszynowych.

Tak duża liczba wad a szczególnie duże marnotrawstwo pamięci dyskowej dosyć mocno ogranicza zastosowanie tego systemu.

6. SOM-51 MUMPS

W latach 1980-81 w Przedsiębiorstwie Systemów Komputerowych Mera-System w Warszawie powstał wielodostępny system operacyjny SOM 51 wraz z interpretacyjnym językiem programowania MUMPS-400. System ten posiadał mechanizmy organizacji i dostępu do hierarchicznej "bazy danych".

Pomimo tego, że w systemie dopuszczalne były rekordy zmiennej długości, to jednak brak jednolitej struktury zbiorów danych oraz standardowych programów obsługi zbiorów znacznie ograniczały jego możliwości. Faktem jest, że ponieważ system nie był rozwijany nie zyskał sobie zwolenników i obecnie najprawdopodobniej nie jest stosowany.

7. SINK

Baza danych SINK została opracowana w Pracowni Projektowej Morskiej Obsługi Radiowej Statków w Gdyni.

Może pracować pod systemem operacyjnym CROOK-3. Nie będę szczegółowo opisywał tej bazy, gdyż Czytelnik znajdzie jej w miarę dokładny opis w materiałach pokonferencyjnych III Konferencji. Wyeksponuję jedynie jej najistotniejsze cechy.

Podstawowym warunkiem zastosowania jest konieczność posiadania jednostki grupowej monitorów ekranowych MERA 7900.

Z końcówki operatora systemu wywoływany jest program główny na monitory jednostki grupowej, który realizuje funkcje zakładania zbiorów, wprowadzania i modyfikacji danych oraz selektywnego wyszukiwania i wyświetlania informacji.

Programy realizujące funkcje zmiany struktury zbiorów, drukowania zestawień, kompresji, jak również użytkowe programy działające na zbiorach mogą być wykonywane tylko i wyłącznie z końcówki operatora systemu.

Olbzrymią zaletą tej bazy jest duża szybkość działania. Jednostka grupowa może pracować z szybkością 9600 a nawet 19200 bodów. Przeglądanie zawartości zbiorów jest wykonywane przez pobieranie do pamięci operacyjnej bloków po 24 sektory /jeden cylinder/. Inną istotną zaletą jest zmienna długość rekordów dzięki czemu istnieje duża efektywność w wykorzystaniu pamięci zewnętrznej. W trakcie wprowadzania danych rekordy logiczne z różnych zbiorów mogą być przemieszane. Każdy rekord logiczny posiada adres identyfikujący jego położenie. Kompresja powoduje pogrupowanie rekordów należących do tych samych zbiorów.

Wydaje się jednak, że brakuje tutaj możliwości sortowania zbiorów, a w wielu użytkowych systemach przetwarzania danych jest to wręcz niezbędne.

WNIOSKI

W tego rodzaju opracowaniu trudno jest precyzyjnie określić wszystkie cechy baz a ich podział na zalety i wady jest często sprawą subiektywną.

Z przedstawionych krótkich charakterystyk wynika, że nie ma w tej chwili bazy danych na MERA 400, która nosiłaby cechy bazy uniwersalnej i mogła być efektywnie wykorzystywana przy dużych

konfiguracjach.

W trakcie spotkania wypracowano następujące wnioski:

1. Jest sens opracowania nowoczesnego, bardziej uniwersalnego systemu zarządzania zbiorami na miarę nowej generacji MERA 400 /duże pojemności pamięci operacyjnych i zewnętrznych/.

2. Określono podstawowe cechy nowej bazy:

- organizacja fizyczna i logiczna powinna obejmować różne struktury, które są używane w aktualnie eksploatowanych bazach,
- możliwość wyboru struktury zbiorów w celu zoptymalizowania procesów przetwarzania w zależności od specyfiki systemu użytkowego ,
- mechanizmy ochrony zbiorów przed upadkiem /możliwość odtwarzania/,
- możliwość łatwego przejścia na nową bazę z najczęściej używanych baz danych np. metodą przegrywania zbiorów przy zachowaniu logicznej struktury i dorobienia mechanizmów do programów użytkowych,
- możliwość wyboru komunikacji przez operatora /język dyrektyw lub konwersacja w języku polskim/,
- dokładna dokumentacja dla początkujących i zaawansowanych na nośniku,
- dopuszczalne pomocnicze dyrektywy objaśniające /tzw. helpy/,
- powinna uwzględniać współpracę z pamięciami dyskowymi typu Winchester,
- język programowania umożliwiający pisanie programów użytkowych działających na zbiorach bazy,
- mechanizmy śledzenia programów w tym języku w trakcie uruchamiania,
- możliwość zastosowania w sieciach z końcówkami inteligentnymi, .
- standardowe formaty danych przy wprowadzaniu,
- maksymalna liczba dokumentów w zbiorze nie powinna być mniejsza od 60 tys.,

3. Zaproponowano zasady opracowania bazy:

- w niektórych etapach tworzenia bazy dopuszcza się dzia-

- łanie zespołów konkurencyjnych,
- przed rozpoczęciem prac należy ustalić wspólne założenia, standardową koncepcję, która będzie obowiązująca dla zespołów,
 - cykliczne spotkania zespołów powinny doprowadzić do realizacji najlepszego projektu,
 - potrzebny jest ośrodek koordynujący prace,
 - konieczne jest ustalenie zasad i form finansowania.

mgr inż. Lech Byczkowski
Morska Obsługa Radiowa Statków
Gdynia

ILUSTRACJA MOZLIWOSCI BAZY DANYCH "SINK"
NA PRZYKŁADZIE OBSŁUGI MIĘDZYNARODOWEGO
TURNIEJU KOSZYKÓWKI KOBIET O "PUCHAR BAŁTYKU"

I. Skrótowy opis.

Wstęp

Technologia baz danych jest współczesnym wyjściem naprzeciw problemom związanym z przechowywaniem, przetwarzaniem i udostępnianiem informacji. Umożliwia stosunkowo szybkie i elastyczne rozwiązywanie rozmaitych zadań. Specjalistyczne oprogramowanie biblioteczne operujące na znormalizowanych strukturach danych skraca czas pisania wymaganych procedur wyszukiwania i przetwarzania oraz uzyskiwania dogodnej postaci przez drukowane dokumenty wynikowe.

Wśród rozróżnianych 3 rodzajów baz danych: hierarchicznych, sieciowych i relacyjnych, te ostatnie pomimo trudności implementacyjnych, spełniają wiele wymagań stawianych systemom baz danych. Przede wszystkim posługują się stosunkowo prostym i łatwym do zrozumienia aparatem pojęciowym znanym z matematyki, informacje przedstawiane są za pomocą jednej struktury danych: relacji, dostępne są w nich operatory wysokiego rzędu. Model relacyjny ma silne podstawy teoretyczne oraz na przestrzeni ostatnich lat nadal inspirowa badania w takich dziedzinach jak współbieżność, blokowanie, ochrona, normalizacja i semantyka danych.

Wymagania sprzętowe

Projektowana w PP MORS na minikomputerze MERA-400 baza danych oparta jest również o pewne elementy modelu relacyjnego. Przy jej opracowywaniu kierowano się optymalnym wykorzystaniem najtańszego i najlepiej nadającego się naszym zdaniem do tych

celów systemu monitorów ekranowych MERA-7900. System ten umożliwia formatowanie ekranu tzn. określanie pól chronionych lub niechronionych dla wprowadzania danych, pól numerycznych i alfanumerycznych, których wypełnianie sprawdzane jest sprzętowo, pól niewyświetlanych, pól wykrywalnych, piórem świetlnym, posiada adresowany ekran, klawisze funkcyjne, które można dowolnie oprogramować, identyfikator operatora, możliwość podłączenia drukarki pracującej jako hard-copy itp.

Poza posiadaniem systemu MERA-7900 nie zakładamy żadnych szczególnych wymagań w stosunku do liczby jednostek pamięci dyskowych czy też pojemności PAO, chociaż oczywiście ich wielkość ma wpływ na efektywność działania bazy danych i ogranicza ją fizycznie.

Założenia projektowe

Dążono do osiągnięcia następujących cech eksploatacyjnych:

- możliwość korzystania z systemu przez użytkowników nieinformatyków,
- możliwość równoległej, bezkonfliktowej pracy wielu użytkowników z tym samym lub różnymi zbiorami danych,
- wzajemna ochrona zbiorów danych poszczególnych użytkowników przed zniszczeniem lub dostępem osób nieupoważnionych,
- wstępna kontrola wprowadzanych danych,
- tryb pracy z tzw. "menu",

oraz technicznych:

- krótki czas reakcji przy pracy on-line,
- oszczędna gospodarka pamięcią dyskową /stosowano głównie jednokierunkowe listy rekordów zmiennej długości/,
- elastyczność i łatwość rozbudowy eksploatowanego systemu,
- nie narzucanie innych ograniczeń niż fizyczne.

Stan bieżący

Użytkownik bazy danych ma możliwość samodzielnego zarejestrowania się podając swe nazwisko oraz hasło. Dane te będą każdorazowo sprawdzane przed rozpoczęciem sesji. Po dokonaniu w/w czynności ma on dostęp do tych zbiorów bazy danych, które mają status ogólnodostępnych. Inni użytkownicy bazy mogą udzielać nowemu abonentowi stałych, ograniczonych lub pełnych upraw-

nień do odczytu, modyfikacji, kasowania i dopisywania nowych rekordów, jak również wycofać lub ograniczyć udzielone uprawnienia. Nowy użytkownik eksploatuje istniejące w bazie zbiory danych lub zakłada własne stając się automatycznie dysponentem uprawnień do wykonywania na nich operacji przez pozostałych użytkowników. Zakładanie nowego zbioru polega na podaniu jego nazwy oraz struktury tj. opisów kolejnych pozycji składających się z nazwy pozycji, jej typu, maksymalnej długości w znakach oraz dodatkowej informacji o tym czy jest to pozycja o stałej długości, czy jej obecność w rekordzie jest obowiązkowa itp. Podane dane służą do kontroli sprzętowej i programowej zawartości pozycji przy jej wprowadzaniu i modyfikacji.

W momencie zarejestrowania powyższej struktury można już wprowadzać, usuwać i modyfikować rekordy nowego zbioru. Może to robić zarówno właściciel zbioru jak i abonenci posiadający nadane im indywidualnie przez właściciela zbioru uprawnienia, również równolegle.

Poza wymienionymi funkcjami użytkownicy mogą odnajdywać rekordy spełniające nałożone na nie warunki, modyfikować wprowadzone warunki wyboru dokumentów lub obliczać liczbę rekordów je spełniających. Rekordy spełniające nałożone warunki mogą być wyświetlane kolejno jeden po drugim lub też użytkownik może zażyczyć sobie jednoczesnego wyświetlenia określonych pozycji różnych rekordów. W tym drugim przypadku abonent otrzymuje na ekranie monitora zadane pozycje wielu rekordów spełniających nałożone warunki, co ułatwia porównywanie ich zawartości i ma duże znaczenie praktyczne, szczególnie przy pracy z większymi zbiorami.

Należy zaznaczyć, że wszystkie wymienione funkcje wykonywane są w trybie tzw. "menu" przez odpowiednio oprogramowane klawisze funkcyjne, a abonent ma do dyspozycji pełne możliwości edycyjne monitorów systemu MERA-7900. System odporny jest na błędy operatorskie, sygnalizując wciśnięcie nieodpowiedniego w danym momencie klawisza funkcyjnego. Abonent ma możliwość wykonania hard-copy informacji zawartej na ekranie monitora.

Poza wymienionymi w dużym skrócie funkcjami systemu w trybie on-line istnieje szereg programów pracujących w trybie wsadowym. Są to m. in. programy kompresji bazy danych oraz pro-

gramy umożliwiające dostęp do zbiorów bazy danych przez programy pisane np. w języku BASIC. Zwiększają one dodatkowo walory użytkowe prezentowanej bazy danych.

Przewidywany rozwój

Obecnie prowadzone są prace nad wprowadzeniem m. in. w ograniczonym zakresie arytmetyki przez co rozumiem:

- możliwość sumowania na bieżąco wybranej pozycji rekordów spełniających zadane warunki /np. w celu określenia sumy wartości materiałów wybranego lub wszystkich magazynów, sumy wartości środków trwałych w przedsiębiorstwie/,
- możliwość automatycznego tworzenia pozycji rekordu w wyniku wykonania operacji arytmetycznych na innych pozycjach tego rekordu /np. w celu określenia wartości danego materiału w magazynie na podstawie jego ceny jednostkowej i stanu magazynowego/, oraz kopiowania do innego zbioru wybranych rekordów zbioru aktualnie przetwarzanego i udoskonaleniem oprogramowania drukującego żądane raporty na podstawie zawartości informacyjnej bazy danych. Wspomniane prace rozwojowe mają na celu zaspokojenie specyficznych potrzeb pojawiających się w systemach obrotu materiałowego, finansowo-księgowych itp.

Wykorzystanie

Przewiduje się wykorzystanie opisywanej bazy danych poza PP MORS w przedsiębiorstwach średniej wielkości przykładowo w zakresie:

- kartoteki kadrowej,
- kartoteki środków trwałych,
- kartoteki obrotu materiałowego,
- systemu fakturowania,

...

Wymienione dziedziny stanowią jedynie przykłady możliwych obszarów zastosowań. Oprogramowanie bazy danych jest w dużej mierze uniwersalne i jak potwierdziło to już doświadczenie może być wykorzystane w zależności od potrzeb i inwencji użytkowników do bardzo szerokiego kręgu zagadnień.

II. Przykład wykorzystania

Informacje wstępne

W maju b.r. w Gdańsku odbyła się ważna nie tylko ze względów sportowych impreza: Międzynarodowy Turniej Koszykówki Kobiet o "Puchar Bałtyku". W zawodach wzięły udział reprezentacje Chin,

Czechosłowacji, Węgier, Rumunii, ZSRR, Szwecji, USA, Jugosławii, Polski I i Polski II. Było to najważniejsze po Igrzyskach Olimpijskich w Los Angeles spotkanie tych zespołów z uwagi na bojkot Olimpiady przez państwa naszego obozu.

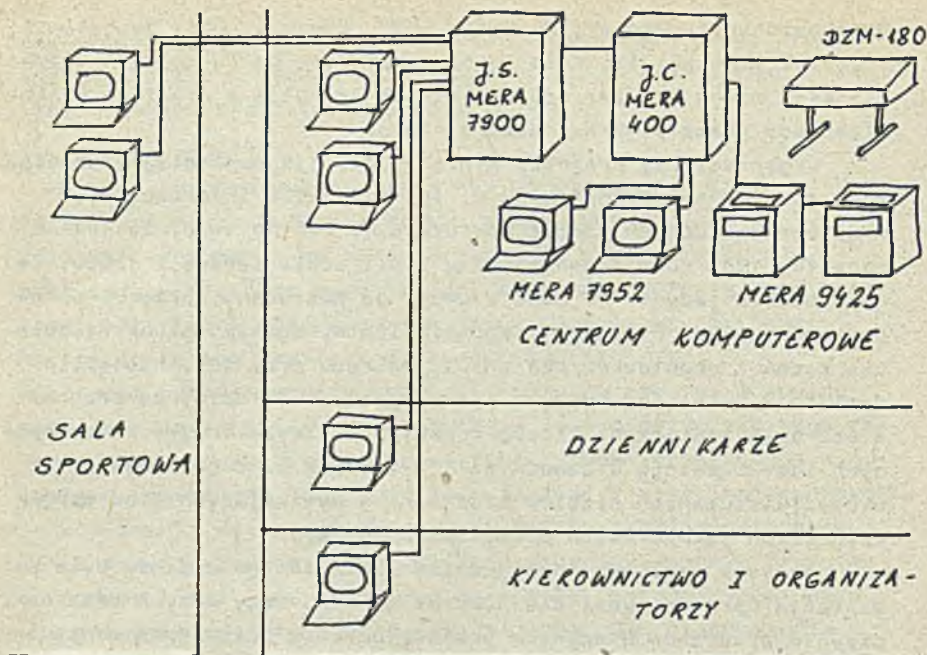
Organizatorzy pragnęli wywiązać się jak najlepiej z obowiązków gospodarza wprowadzając m. in. dla celów informacyjnych oraz obróbki danych sprzęt komputerowy. Pomimo tego, że już na początku 1985 roku organizatorzy zdali sobie sprawę z faktu, że na Wybrzeżu jedynie PP MORS z uwagi na posiadany sprzęt i oprogramowanie jest w stanie zaspokoić ich wymagania, skonkretyzowanie żądań i obowiązków obu stron, zakresu prac itp. nastąpiło dopiero w kwietniu. Muszę w tym miejscu stwierdzić, że programiści byli zaskoczeni liczbą rozmaitych elementów gry obserwowanych indywidualnie u zawodniczek w trakcie każdego meczu przez wyspecjalizowanych sędziów oraz liczbą wynikających stąd współczynników jakościowych oceny zawodniczek.

Jedynym możliwym rozwiązaniem postawionego zadania była adaptacja opracowywanej dla potrzeb zakładu bazy danych oraz napisanie programów wsadowych pobierających z bazy odpowiednie informacje i drukujące na ich podstawie raporty zamówione przez organizatorów.

Realizacja sprzętowo-programowa

Zaprojektowano i zrealizowano następującą konfigurację sprzętową /rys. 1/: 2 monitory MERA-7910 postawiono w sali, na której rozgrywano zawody, 1 w pomieszczeniu zajmowanym przez dziennikarzy i 1 w pokoju kierownictwa i organizatorów zawodów. W centrum komputerowym stała jednostka centralna MERA-400, 2 pamięci dyskowe MERA-9425, 2 monitory operatorskie MERA-7952, drukarka DZM-180, jednostka sterowania grupowego MERA-7905 i 2 monitory zależne MERA-7910.

Ze strony programowej uproszczono dialog na monitorach stojących poza centrum komputerowym oraz ograniczono ich funkcje do możliwości uzyskiwania informacji ze zbiorów związanych z zawodami. Monitory stojące w centrum nie były ograniczone funkcjonalnie i służyły do wprowadzania nowych informacji, ewentualnie modyfikacji, czy poprawiania istniejących. Przy ich pomocy zapoznawano również niektórych gości z funkcjonowaniem i obsługą systemu oraz udzielano dziennikarzom bardziej wyrafinowanych informacji przekrojowych, których uzyskanie wymagało pewnej biegłości w obsłu-



Rys. 1 Rozmieszczenie sprzętu komputerowego

dze.

Informacje dotyczące zawodów gromadzone były w 5 zbiorach:
SEDZIORWIE - zawierającym imię i nazwisko sędziego, narodowość, miejsce zamieszkania, rok urodzenia, zawód, staż sędziowski i staż FIBA,

EKIPY - zawierającym nazwę państwa, nazwiska trenerów, kierownika ekipy, skład zespołu, średnią wzrostu, średnią wieku, ważniejsze osiągnięcia sportowe itd.

ZAWODNICZKI - zawierającym nr startowy, imię i nazwisko zawodniczki, narodowość, wzrost, wiek, nazwę klubu oraz czas gry w reprezentacji,

STRZELCY - zawierającym pozycję w klasyfikacji, nr startowy i nazwisko zawodniczki, narodowość, liczbę zdobytych punktów, łączny czas gry, średnią liczbę punktów zdobywanych w meczu ...

OBSERWACJE - gromadzącym informacje o grze każdej zawodniczki w każdym spotkaniu tzn. m. in. nr zawodniczki, nazwisko, narodowość, mecz zespołów, datę meczu, wynik, sędziujących oraz odpowiednio w 1. i 2. połowie: liczbę rzutów za 2 pkt,

liczbę celnych rzutów za 2 pkt, podobnie za 3 pkt i rzuty wolne, liczbę zdobytych pkt, zbiórki w obronie, zbiórki w ataku, przechwyty piłki, asysty, straty, faule, czas gry. Największymi zbiorami były: zbiór zawodniczek obejmujący 118 rekordów i zbiór obserwacji obejmujący po zakończeniu zawodów 629 rekordów zajmujących ponad 10 cylindrów.

Po zakończeniu każdego meczu i wprowadzeniu do systemu związanych z nim danych drukowano dla każdej reprezentacji 2 raporty. Pierwszy tzw. arkusz obserwacji zawierał informacje pobrane ze zbioru OBSERWACJE dotyczące indywidualnej postawy każdej zawodniczki w danym spotkaniu, dodatkowo obliczone wskaźniki procentowe i informacje uśrednione dla całej drużyny.

Drugi tzw. arkusz efektywności zawierał 7 wskaźników liczbowych, tzw. wskaźników efektywności gry Valkowa oceniających poszczególne elementy gry każdej zawodniczki w danym spotkaniu, wskaźnik sumaryczny oraz informacje uśrednione dla całej drużyny.

Te 2 dokumenty stanowią dla trenerów podstawowy materiał indywidualnej oceny zawodniczek i interesowały zarówno bezpośrednich uczestników meczu, jak i prowadzących pozostałe zespoły. W założeniu miały być dostarczane zainteresowanym najpóźniej następnego dnia rano, aby przed kolejnymi spotkaniami trener miał możliwość reorganizacji zespołu czy też zwrócenia uwagi na pewne elementy gry w czasie treningów. Udało się to zrealizować dopiero w końcowych dniach turnieju z 2 powodów. Pierwszym, ale nie najważniejszym były kłopoty sprzętowe. Drugim opóźnienie dostarczenia danych do centrum komputerowego oraz wewnętrzne sprzeczności w dostarczanych danych. Okazało się, że osoby prowadzące protokoły obserwacji nie były przyzwyczajone do rygorystycznego przestrzegania zasad narzucanych przez komputerowy system przetwarzania. Dotychczas stosowane metody obliczeń nie ułatwiały ujawniania wewnętrznej sprzeczności danych, autorzy oprogramowania uważali zaś za swój obowiązek prowadzenie tak głębokiej weryfikacji jak było to tylko możliwe.

Poza wymienionymi raportami w końcowych dniach turnieju drukowano również aktualne listy rankingowe zawodniczek wg różnych kryteriów, a mianowicie:

- najlepszych strzelców,
- najlepszych w rzutach za 3 pkt,
- najlepszych w rzutach wolnych,
- najlepszych w zbiórkach z tablicy.

Podstawę klasyfikacji stanowiły dane zawarte w zbiorze OBSERWACJE.

Na zakończenie turnieju wydrukowano również raporty końcowe dla wszystkich drużyn podające łączne dane o indywidualnej postawie zawodniczek we wszystkich spotkaniach, uzupełnione o podstawowe dane dotyczące zawodniczki takie jak wiek, wzrost i liczbę spotkań, w których brała udział. Zamieszczono tam także wartości średnie dotyczące całej drużyny.

Ocena działania

Przy realizacji systemu opierano się na rozwiązaniu przyjętym przez węgierską firmę "Videoton" na Mistrzostwach Europy w Koszykówce Kobiet rozegranych w Budapeszcie w roku 1983. Wyeliminowano znane niedogodności i wprowadzono elementy zaproponowane przez organizatorów. System spełnił ich oczekiwania. Należy jednak skrytykować tak późne przystąpienie do realizacji systemu dla tak prestiżowej imprezy. Niewątpliwie należało przeprowadzić próby obsługi kilku meczy drużyn krajowych, co ułatwiłoby zrozumienie wzajemnych obowiązków organizatorów i wykonawców oraz wyeliminowało opóźniony start na początku turnieju.

Doświadczenia i wnioski

Opisywane oprogramowanie bazy danych potwierdziło słuszność przyjętych założeń projektowych. Okazało się wystarczająco elastyczne dla szybkiej adaptacji do obsługi imprezy nie leżącej w obszarze zainteresowań autorów.

W przyszłości już bez obaw możemy podjąć się obsługi podobnych imprez sportowych, czy handlowych w naszym regionie. Należy wspomnieć, że opisywana baza danych jest intensywnie eksploatowana w PP MORS przez 6 godzin dziennie na monitorach rozproszonych po różnych komórkach organizacyjnych, u bezpośrednich użytkowników reprezentujących różne zawody i poziomy wykształcenia. Nie wymaga ona żadnych interwencji operatora systemu, a czasy reakcji są rzędu ułamków sekund.

Na zakończenie jeszcze jedna uwaga programisty: żaden ze znanych nam systemów operacyjnych nie posiada efektywnych ekstrakodów do pełnej i dobrej jakościowo obsługi jednostki grupowej. W związku z tym w PP MORS opracowano własny pseudohandler, wykorzystujący naszym zdaniem w sposób najbardziej efektywny jej możliwości.

Wykonawcy

Obsługę Międzynarodowego Turnieju Koszykówki Kobiet o "Puchar Bałtyku" przeprowadził zespół PP MORS w składzie:

M. Szczepański, Z. Krawczyk - strona sprzętowa,

R. Piechota, J. Ługiewicz, L. Byczkowski - strona programowa.

Wykonawcy składają serdeczne podziękowanie trenerowi K.S. "Spójnia" mgr Kazimierzowi Wierzbickiemu reprezentującemu organizatorów od strony merytorycznej, bez którego uwag i współpracy realizacja systemu byłaby niemożliwa. /

mgr Maria Meler-Kapcia
Instytut Okrętowy
Politechniki Gdańskiej

ROZWÓJ ZASTOSOWAŃ BAZY DANYCH SELKO

Baza danych SELKO opracowana z przeznaczeniem do fakturowania usług stoczni remontowej i w pełni wdrożona sprawdziła się również w zakresie innych zastosowań. Aktualnie funkcjonuje w księgowości Instytutu Okrętowego do prowadzenia rozliczeń i kontroli wykonania umów BZ w zakresie planowania i bieżącego ich wykorzystania. Służy również ewidencji przychodów i rozchodów aparatury w instytucie.

W bieżącym roku wykorzystana została do sporządzania wszelkich dokumentów wymaganych w systemie rekrutacji na studia w IO. SELKO znalazła również zastosowanie w dziekanacie tegoż instytutu do obliczania oraz sporządzania listy wypłat stypendiów z uwzględnieniem wszelkich dopłat i potrąceń.

Równolegle prowadzona będzie ewidencja wyników nauczania studentów wszystkich lat, co umożliwi dodatkowo obliczanie premii za naukę.

W przyszłym roku przewidywane jest rozszerzenie zakresu zastosowań o finansowanie innych sfer działalności instytutu. Ten dość różnorodny przebieg użytkowania bazy danych SELKO pozwolił na dokonanie pewnego podsumowania, a mianowicie:

- Potwierdzone zostały wszechstronne możliwości w zakresie rozwiązywania praktycznych zastosowań bez potrzeby pisania pomocniczych programów użytkowych poza obzarem systemu bazy danych.

- Ustalony został standardowy schemat przetwarzania dla wszystkich zastosowań obejmujący kolejno: zakładanie zbiorów danych, zapisywanie i aktualizację danych, sortowanie zbiorów, ewentualne przebiegi programu JOIN dla złożonego przetwarzania i obliczeń oraz wydruki bieżące /głównie w okresie testowym/ i wydruki standardowe /po uzgodnieniu wszystkich szczegółów z użytkownikiem/.
- Sprawdziły się praktyczne walory makroprocedur nie tylko ogólnego przeznaczenia /makro DRUK/, ale w odniesieniu do poszczególnych zadań użytkowych jak np. obliczania stypendium, ustalania ocen i tzw. małych punktów z egzaminów wstępnych czy wreszcie rozliczenia i zestawienia funduszu honorariów planowanego i wykonanego za dowolny okres.
Makroprocedury umożliwiają tworzenie specjalizowanych pakietów użytkowych dla określonych zastosowań bez konieczności ingerencji użytkownika w proces przetwarzania. Mogą one obejmować kolejne odwołania do wielu programów realizowanych w wielu przebiegach. Stanowią ciąg makrozleceń wykonywany w pełni automatycznie bądź parametryzowany z możliwością podawania na bieżąco wymaganych parametrów aktualnych. Po dopracowaniu szczegółowego trybu przetwarzania możliwe jest stworzenie makroprocedur dla poszczególnych zadań użytkowych w miejsce konwersacji. Trzon makroprocedury stanowi ciąg odpowiedzi użytkownika w programach bazy danych.
- Nadal istnieją ograniczenia co do wielkości zbiorów ze względu na czas ich przetwarzania. Z uwagi na to, że średni czas przetwarzania jednego rekordu waha się w granicach 0.1 - 0.5 sek, długość zbioru danych nie powinna przekraczać kilku tysięcy rekordów.

- Niezmiernie użyteczny okazał się walor elastyczności bazy danych ze względu na bardzo częste zmiany przepisów i ustaleń ze strony użytkowników.

Niezbędne reorganizacje zbiorów były łatwe do przeprowadzenia, a zadania do redefiniowania.

Zrealizowane dotąd oprogramowanie bazy danych SELKO umożliwia przetwarzanie struktur płaskich bez możliwości odwzorowania wzajemnych powiązań między zbiorami za pośrednictwem kluczy obcych i relacji funkcjonalnych. W dotychczasowych zastosowaniach nie miało to większego znaczenia, natomiast niezbędne okazało się w sferze zastosowań technicznych do opisu obiektów złożonych jak np.: samochód, statek.

Istnieje zatem praktyczna potrzeba odwzorowania w relacyjnej bazie danych struktur hierarchicznych, a niekiedy sieciowych.

Przedsięwzięcie takie wymagałoby:

- uwzględnienia reprezentacji wszystkich typów kluczy takich jak: główny, potencjalny, obcy,
- oprogramowania algorytmu tworzenia ścieżki dostępu do wielu zbiorów umożliwiającego nawigację w bazie danych,
- interpretacji nazwy zbioru jako danej,
- proceduralnej struktury programów systemu bazy danych,
- szybkich metod dostępu do zbiorów.

Pierwsze próby takich prac/dot. pktów 1 i 2/ zostały wykonane łącznie z testowaniem programów .

Poegały one na automatycznym generowaniu ścieżki dostępu do wszystkich zbiorów wymaganych przez zadanie. Za podstawę realizacji przyjęto operator łączenia /JOIN/ oparty na równości. Aby użytkownik nie musiał orientować się w strukturze zbiorów bazy danych przewidziano swoistą nadbudowę dla programu JOIN,

umożliwiająca posługiwanie się wyłącznie nazwami pól, które użytkownikowi kojarzą się z konkretnymi zastosowaniami.

Algorytm skonstruowany do tego celu uwzględnia dwa etapy realizacji:

- w pierwszym etapie na podstawie nazw pól wyszczególnionych dla dokonania żądanej operacji w b.d. i informacji o kluczach ustala logiczną ścieżkę dostępu do zbiorów. Umieszcza opisy wszystkich zbiorów, do których należą podane pola, eliminując następnie opisy nie posiadające powiązań z innymi,
- w drugim etapie tworzy zbiór roboczy, będący kompozycją zbiorów wyselekcjonowanych w pierwszym etapie poprzez wieloprzebiegowe łączenie.

W odniesieniu do tego zbioru możliwe są wszelkie operacje systemu bazy danych.

Wykonane dotychczas prace w kierunku reprezentacji struktur hierarchicznych związane są z wersją modelową systemu.

Reasumując:

- Baza danych SELKO może znaleźć zastosowanie wszędzie, gdzie wielkość zbiorów nie przekracza kilku tysięcy rekordów z uwagi na fakt, że oprogramowanie w większej części napisane zostało w języku BASIC.
- Jest szczególnie użyteczna tam, gdzie istnieje duża różnorodność zastosowań i częste zmiany sposobu przetwarzania.
- Nie wymaga pisania programów użytkowych, co stanowi jej niewątpliwą zaletę.
- W przypadku przetwarzania rutynowego mogą być tworzone makrobloki, eliminujące uciążliwą na tym etapie konwersację.

- Dla użytkowników, którzy chcą uzyskać krótszy czas przetwarzania istnieje możliwość wprowadzenia dla zadań, których czas jest krytyczny, procedur języka RAL, kompatybilnego z bazą danych SELKO.

mgr inż. Maria Perek
mgr inż. Jan Wierzbicki
Huta Szkła Okiennego
"Szczakowa"

ZASTOSOWANIE BAZY DANYCH BD-83 VITRIN
POD SYSTEMEM OPERACYJNYM CROOK-4

W związku z opracowaniem w ostatnich dwóch latach nowych, o wiele bardziej efektywnych systemów operacyjnych pojawił się dla użytkowników standartowego SOM'a problem przejścia na te systemy. O ile w przypadkach systemów somopodobnych problem ten był mało istotny, to jeśli chodzi o CROOK'a-4 nabrał on poważnego znaczenia.

Reprezentujemy kilka Ośrodków, które łączą wspólne używanie Bazy Danych BD-83 Vitrin. Ponieważ Ośrodki te zdecydowały się przejść na CROOK'a-4 konieczne było w pierwszym rzędzie przeniesienie właśnie tego narzędzia przetwarzania. Dzięki wspólnym wysiłkom w zeszłym miesiącu zostało zakończone przerabianie Bazy na CROOK'a. W Ośrodkach, które będą stosowały oprogramowanie Bazy Danych BD-83 pod systemem operacyjnym CROOK'4 obecnie jest eksploatowanych wiele systemów użytkowych z wykorzystaniem tej bazy pod SOM-em, dlatego projektując wersję bazy pod CROOK'iem-4 starano się zachować bez zmian wszystkie funkcje i rozwiązania aby zapewnić możliwie łagodne przejście z jednego systemu operacyjnego na drugi.

W zasadzie dokumentacja systemu:

A. Ziółkowski: "Oprogramowanie Bazy Danych BD-83 dla minikomputera Mera-400", Tarnobrzeg 1983.

obowiązuje również dla wersji oprogramowania BD-83 pod systemem operacyjnym CROOK'4. Najpoważniejsze zmiany wprowadzono w programie DANE. Ze względu na dynamiczny przydział pamięci operacyjnej i wielozadaniowość w systemie CROOK-4 nie ma specjalnego uzasadnienia dla wielodostępnej wersji programu DANE. Rezygnując z wielodostępu w programie tym wprowadzono nowe funkcje a mianowicie kontrolę zakresu wprowadzanych liczb oraz możliwość kopiowania pól z ostatnio wprowadzonego rekordu.

W oprogramowaniu Bazy Danych BD-83 pod systemem operacyjnym SOM-3 zbiory danych są przechowywane na sekcjach przy czym jedna sekcja może zawierać wiele zbiorów danych. W wersji pod systemem CROOK-4

zastąpiono sekcję zbiorem systemu CROOK-4. Zbiór ten podobnie jak sekcja może zawierać wiele zbiorów bazy danych BD-83. Przy wyświetlaniu formatu informacji do wprowadzania zachowano nazwę "sekcja". Zachowano również ograniczenie długości nazwy "sekcji" wynoszące 3 znaki, tak więc nazwy zbiorów systemu CROOK-4, w których przechowywane są zbiory bazy danych BD-83 mogą mieć co najwyżej 3 znakowe nazwy. We wszystkich programach bazy danych zlikwidowano wywołanie ekstrakodów systemu operacyjnego SOM-3: CRE i REM. Ekstrakod ASS zastąpiono odpowiednikiem z systemu operacyjnego CROOK-4 zachowując formę wywołania.

Pewne problemy stwarza przyjęte w FORTRANIE-CROOK ograniczenie, że numery strumieni przywidywanych do zbiorów bezpośredniego dostępu mogą przyjmować wartości z zakresu 30 do 50.

W dotychczas pisanych programach użytkowych stosowano zazwyczaj niższe numery strumieni z zakresu 1 do 20. Aby uniknąć wprowadzania poprawek w wielu miejscach, w dotychczas napisanych programach dopuszczono używanie strumieni z zakresu 1 do 20 przy wywoływaniu procedur bazy danych BD-83. Numery te są automatycznie zwiększane o 30, natomiast numery strumieni z zakresu 30 do 50 są pozostawiane bez zmian. W programach użytkowych korzystających ze zbiorów bezpośredniego dostępu należy dyrektywę sterującą kompilacją #DEF. W dyrektywie tej należy podać numery strumieni z zakresu 30 do 50. /np. Jeżeli w procedurach używa się strumienia o numerze 4 w #DEF należy podawać numer 34/.

Podobnie jak pod systemem SOM-3 uruchamiając programy użytkowe można korzystać z wydruków pomocniczych drukowanych na strumieniu nr 9. System CROOK-4 umożliwia zapisywanie tych wydruków w zbiorach dyskowych. Ponieważ teksty są zapisywane w postaci upakowanej/a nie jak w systemie SOM-3 1 rekord na sektor/ należy polecić ten sposób korzystania z wydruków pomocniczych.

Jak już wcześniej wspomniano pod systemem CROOK-4 program DANE zrealizowano w wersji jednodostępnej. Program realizuje wszystkie dyrektywy opisane w dokumentacji. Ponadto w programie wprowadzono

- możliwość kopiowania pól ostatnio wprowadzonego/ lub wyświetlanego/ rekordu,
- kontrolę zakresu wprowadzanych danych.

Inaczej niż w poprzedniej wersji, program DANE przy wprowadzaniu każdego pola, wyświetla wartość tego pola w ostatnio wprowadzonym lub wyświetlanym rekordzie /zamiast znaków# określających długość

pola. Użytkownik może wprowadzić wyświetloną wartość naciskając klawisz CR lub wprowadzić z klawiatury nową wartość pola. Przy powtarzających się wartościach danych w wielu kolejnych rekordach wprowadzona modyfikacja pozwala oszczędzić czas na wprowadzenie tych danych.

Kontrolę zakresu wprowadzanych danych numerycznych zrealizowano wprowadzając do każdego zbioru 2 rekordy: rekord ograniczeń dolnych oraz rekord ograniczeń górnych. Znaki * oznaczają brak ograniczeń. Rekordy te są zapisane jako przedostatni i ostatni w zbiorze. Oznacza to, że zbiór może zawierać co najwyżej N-2 rekordy danych gdzie N-maksymalna liczba rekordów w zbiorze zadeklarowana programem BAZA. Powyższe rozwiązanie, niezbyt może eleganckie, ma jednak tę zaletę, że zachowuje dotychczasową budowę zbiorów.

Od niedawna w naszym Ośrodku rozpoczęliśmy pod CR00K'iem i bazą HD-83 eksploatację pierwszego przerobionego systemu rozliczania produkcji. System ten rozlicza wykonanie produkcji szkła w poszczególnych fazach procesu technologicznego, drukując raporty dobowe, rozliczenia miesięczne jak również obliczanie akordów grup pracowników bezpośrednio produkcyjnych. Rozpoczęliśmy również eksploatację systemu "TECHNOLOGIA", prowadzącego ewidencję analiz technologicznych, obliczającego parametry fizykochemiczne szkła oraz korekty naważek surowców. W systemie tym jest możliwe również drukowanie krzywych lepkości dla masy szklanej.

Powyższe systemy eksploatujemy pod CR00K'iem od miesiąca. Obecnie podstawowym problemem jest wolne działanie programów przetwarzających oraz bardzo istotna sprawa - gubienie danych przy ich wprowadzaniu. Polecamy to szczególnej uwadze autorom CR00K'a-4.

IV SYSTEMY APLIKACYJNE

mgr inż. Tadeusz DYRGA
Instytut Elektroenergetyki i Automatyki
Politechniki Gdańskiej

SYSTEM FINANSOWO - KOSZTOWY NA MINIKOMPUTERZE

MERA - 400

Opracowana wersja systemu finansowo-kosztowego /FK/ bazuje na koncepcji opracowanej w Gdańskich Zakładach Nawozów Fosforowych.

System obejmuje zagadnienia ewidencji obrotów i stanów księgowych oraz rozliczenia kosztów.

A. Ewidencja obrotów i stanów księgowych.

Dane źródłowe dotyczące obrotów księgowych wprowadzane są do systemu bezpośrednio z dokumentów księgowych istniejących w przedsiębiorstwie /dowody księgowe/. Dane wprowadzane z dowodów księgowych dotyczą obrotów na poszczególnych kontach oraz łączną sumę tych obrotów w ramach danego dowodu księgowego /suma kontrolna/. W trakcie wprowadzania dane podlegają kontroli, a wykryte nieprawidłowości mogą być natychmiast usunięte. Po zakończeniu wprowadzania danych /wszystkich lub ich części/ system dokonuje kontroli poprawności danych w zakresie dowodów księgowych. W ramach każdego dowodu księgowego następuje sprawdzenie zgodności łącznej sumy obrotów księgowych po stronie WINIEN oraz MA, a także sprawdzeniu zgodności łącznej sumy obrotów księgowych z sumą kontrolną danego dowodu.

W systemie prowadzona jest ewidencja stanów dla poszczególnych kont księgowych narastająco od początku roku. Wydawnictwo ewidencji stanów księgowych obejmuje ewidencję kont analitycznych oraz ewidencję kont syntetycznych na zadane dłu-

gości konta.

W systemie dopuszcza się 12-sto znakowy numer konta księgowego oraz 5-cio znakowy numer dowodu księgowego.

3. Rozliczanie kosztów.

System FK umożliwia rozliczanie kosztów w sposób automatyczny. Rozliczeniu podlegają salda sumy obrotów występujących na wybranych kontach syntetycznych. Rozliczenie kosztów odbywa się przez automatyczne wygenerowanie obrotów księgowych rozliczających. Po zakończeniu rozliczania wygenerowane obroty zostaną włączone do ewidencji obrotów księgowych systemu.

Dopuszcza się stosowanie następujących metod rozliczania kosztów:

- całe saldo występujące na wskazanym koncie lub określona jego część /% / zostaje zapisane na wskazane pary kont,
- całe saldo występujące na wskazanym koncie zostaje rozliczone i zapisane na wskazane konta rozliczające proporcjonalnie do ilości jednostek naturalnych przypisanych do poszczególnych kont rozliczających,
- całe saldo występujące na wskazanym koncie zostaje rozliczone i zapisane na wskazane konta rozliczające proporcjonalnie do liczby procent przypisanych do poszczególnych kont rozliczających,
- całe saldo występujące na wskazanym koncie zostaje rozliczone i zapisane na wskazane konta rozliczające proporcjonalnie do sum obrotów /po stronie winien / występujących na kontach przypisanych do poszczególnych kont rozliczających.

System został zrealizowany na minikomputerze MERA-400 pod systemem operacyjnym CROOK-4. Wymagana konfiguracja minikomputera MERA-400:

- jednostka centralna,
- pamięć operacyjna 64k,
- jednostka pamięci dyskowej,
- 2 jednostki pamięci taśmowej,
- monitor ekranowy z klawiaturą,
- drukarka wierszowa.

Podstawowe zbiory systemu to:

- zbiór ewidencji obrotów księgowych,
- zbiór ewidencji obrotów i stanów księgowych,
- zbiór opisu automatycznego rozliczania kosztów.

W procesie przetwarzania danych w systemie można wyróżnić następujące jednostki funkcjonalne:

- wprowadzanie, aktualizacja i kontrola obrotów księgowych,
- wprowadzanie, aktualizacja i kontrola opisu automatycznego rozliczania kosztów,
- ewidencja stanów księgowych,
- wydawnictwo ewidencji obrotów i stanów księgowych.

Współpraca z użytkownikiem systemu odbywa się w sposób konwersacyjny za pośrednictwem monitora ekranowego z klawiaturą.

W oprogramowaniu systemu zastosowano dwa uniwersalne moduły przetwarzające: WARS oraz SORT.

Moduł WARS służy do wprowadzania danych z terminalu do zbiorów dyskowych systemu. W trakcie wprowadzania dane podlegają kontroli formalnej oraz logicznej. Kontrola obejmuje również relacje zachodzące między poszczególnymi polami rekordu. Dane już zapisane

w zbiorze dyskowym mogą być aktualizowane, poprzez wymianę wskazanych pól rekordów, lub kasowanie zawartości całych rekordów. W trakcie wprowadzania i aktualizacji danych można korzystać z szeregu rozwiązań przyspieszających i ułatwiających proces wprowadzania danych do zbioru. Do rozwiązań tych należą:

- powielanie podanej zawartości pola /całego lub części/ w kolejnych wprowadzanych rekordach,
- kopiowanie wybranych fragmentów pól z rekordu wprowadzonego poprzednio,
- uzupełnienie niepełnej zawartości pola dowolnie ustalonym znakiem z lewej lub prawej strony,
- pomijanie wprowadzania zawartości ciągu pól, których wypełnienie nie jest konieczne,
- aktualizacja zawartości pól rekordu, nie spełniających kontrolowanych relacji między polami, bez konieczności ponownego wprowadzania zawartości rekordu,
- przerwanie wprowadzania danych do rekordu.

Moduł zapewnia również możliwość wyświetlania dowolnie wybranych rekordów lub ciągów rekordów zbioru. Określenie ciągu rekordów odbywa się poprzez podanie zakresów w jakich powinny się zawierać wybrane pola rekordów /lub ich części/.

Wybrane ciągi rekordów mogą być zarejestrowane w specjalnym zbiorze wydawniczym. Zawartość tego zbioru można przesłać na dowolne wyjściowe urządzenie zewnętrzne.

Współpraca z użytkownikiem modułu odbywa się konwersacyjnie za pośrednictwem monitora ekranowego z klawiaturą. W trakcie konwersacji, w dowolnym jej punkcie użytkownik może uzyskać informacje na temat dalszych działań, jakie może podjąć, bez konieczności odwoływania się do dokumentacji eksploatacyjnej.

modułu.

Kodul SORT służy do szybkiego sortowania dowolnego zbioru o dostępie bezpośrednim według zadanych kluczy sortowania. Charakterystyka kluczy sortowania zapisana jest w oddzielnym zbiorze o nazwie określonej przez użytkownika modułu. Zarówno liczba kluczy sortowania jak ich długość może być dowolna. W wyniku sortowania powstaje zbiór adresów rekordów posortowanych. Nazwę tego zbioru określa użytkownik modułu.

Całość oprogramowania systemu wykonano w języku RATFOR
/Rational Fortran/.

SYSTEM PROFIN - PLANOWANIE I ROZLICZANIE FINANSOWE
PRZEDSIĘBIORSTWA

Optymalną strategię przedsiębiorstwa można rozpatrywać z punktu widzenia procesu produkcyjnego i jego fizycznych wyników, bądź też z punktu widzenia polityki finansowej, a dokładniej polityki płacowej. Drugie z tych podejść aczkolwiek mniej właściwe, stało się koniecznością chwili ze względu na trudności materiałowe, remontowe i niestabilizowany rynek. Podstawą systemu komputerowego PROFIN jest model skonstruowany przy takim właśnie podejściu; postawiono sobie za cel jego użyteczność bezpośrednio w przedsiębiorstwie, na bazie informacji z własnej sprawozdawczości finansowej, a więc łatwo dostępnych. Model nie ingeruje w proces produkcyjny, operując jedynie informacjami wynikowymi, w zasadzie kosztowymi. Podstawą do budowy modelu i jego realizacji komputerowej jest schemat gospodarki finansowej przedsiębiorstwa, stale aktualizowany w miarę wprowadzania zmian /i to dość częstych/ aktów prawnych jak np. zastąpienie od roku 1986 obciążeń na PPAZ podatkiem od ponadnormatywnych wypłat. Reforma gospodarcza poszerzyła samodzielność ekonomiczną przedsiębiorstw, przysparzając jednocześnie kłopotów związanych z wyborem strategii finansowej - zbyt dużo istnieje elementów decyzyjnych, by najkorzystniejszy dla przedsiębiorstwa i jego załogi wariant polityki finansowej można było określić intuicyjnie.

Potrzebne więc jest narzędzie umożliwiające bardzo szybki przegląd możliwych wariantów, zaproponowanych w przedsiębiorstwie przez dyrekcję czy czynniki społeczne lub wskazanych jako najlepsze w danych warunkach przez obiektywnego specjalistę. Takimi przeliczeniami zajmuje się zwykle dział księgowości i - jak występuje potrzeba - dostarcza kilku przeliczonych wariantów polityki finansowej. Wybór jest niewielki a możliwość znalezienia najlepszego w danych warunkach wariantu - prawie żadna.

System PROFIN umożliwia przejrzenie kilkudziesięciu takich wariantów, różniących się poszczególnymi elementami kosztowymi i w każdej sytuacji znaleźć rozwiązanie optymalne. Dzięki wykorzystaniu konwersacyjnego trybu pracy, wpływ zaproponowanych zmian na sytuację finansową przedsiębiorstwa widoczny jest już w kilka chwil po ich wprowadzeniu do komputera. Różne wymagania stawiane systemowi takiemu jak PROFIN doprowadziły do określenia trzech trybów pracy. Pierwszy polega na możliwości zasymulowania obliczeń finansowych na koniec okresu /kwartał, półrocze, rok/ ; tryb ten pozwala na przetestowanie hipotetycznych wariantów, które planuje się dużo wcześniej i których skutki dobrze jest przebadać.

Drugi zakłada, że dane /czy też zaplanowane/ są wszystkie wielkości wynikowe działalności gospodarczej poza wynagrodzeniami oraz sumą nagród płacowych z zysku /czyli tzw. "czternastki"/, które mają być ustalone na takim poziomie, by nie powodować żadnych obciążeń na PFAZ.

trzeci idzie jeszcze dalej - umożliwia optymalizację ; wielkościami zmiennymi są nadal wynagrodzenia i nagrody z zysku, zwane dalej wypłatami, funkcją celu zaś może być:
1. suma wypłat, tj. wynagrodzeń i nagród płacowych z zysku,

2. suma odpisów na PFAZ, podatku dochodowego, podatku od płać i ubezpieczeń społecznych,
3. opłacenie każdej złotówki wypłat - podatkami /iloraz funkcji 2 przez 1/, przy czym funkcje 2 i 3 są oczywiście minimalizowane a 1 - maksymalizowana/ minimalizacja funkcji 2 równoważna jest maksymalizacji środków pozostających w przedsiębiorstwie/.

Ograniczeniem naturalnym takiego modelu jest wystarczalność posiadanych środków i jeśli nie nakłada się dodatkowych - jest to optymalizacja bez ograniczeń. Ponieważ wynik finansowy jest miernikiem oceny funkcjonowania przedsiębiorstwa, przewidziano możliwość podania minimalnego dopuszczalnego poziomu wyniku finansowego, jako ograniczenia systemowego. System istnieje już drugi rok i poza zmianami obiektywnymi wynikającymi z nowych przepisów prawnych, przeszedł pewną ewolucję związaną z doświadczeniami uzyskanymi w trakcie eksploatacji systemu.

Ostatnio wzbogacono model o możliwość edycji wyników w formie wykresów decyzyjnych. Jaka była ich geneza ?

Wynikiem optymalizacji czy też prostych przeliczeń jest tabela zbiorcza, przedstawiająca w sposób zwięzły sytuację przedsiębiorstwa, uzupełniona o ewentualne komentarze tekstowe tej sytuacji. Jeśli przetestowano kilka wariantów planu czy też optymalizowano różne funkcje, tabel tych było kilka, kilkanaście czy kilkadziesiąt i wielokryteria-
lne ich porównywanie było niemal niemożliwe. Wykres jest więc bardziej komunikatywnym przedstawieniem wyników w formie graficznej. Zmienną niezależną, określaną, automatycznie przez system jest wartość wynagrodzeń, zależną zaś czyli wyliczaną w trakcie pracy komputera - wynagrodzenia + nagrody lub podatki + PFAZ, bądź w końcu opłacenie wypłat podatkami. Jeden wykres stanowi przegląd wszystkich

wariantów płacowych przy stałych kosztach pozostałych, dla konkretnej funkcji optymalizacyjnej. Dla każdego wariantu podano wartość wynagrodzeń, nagród z zysku, obciążeń /podatki + PFAZ/, wynik finansowy i rezerwę zysku do podziału /po dokonaniu wszystkich odpisów obowiązkowych oraz niezbędnego zadeklarowanego odpisu na fundusz rozwoju/.

Wykres taki jest więc rozszerzeniem możliwości optymalizacyjnych modelu, jako że oprócz wariantu najlepszego można zeń odczytać rozwiązania nieco gorsze ale posiadające inne, ważne dla decydenta zalety. Wykres jest szczególnie przydatny przy podejmowaniu decyzji taktycznych, gdy nie określa się jeszcze konkretnego wariantu, a raczej ich zbiór /przedział/, w którym - po jego wyborze - dokonywane są bardziej szczegółowe obliczenia. Lewa kolumna /zmienna niezależna/ podaje wartość wynagrodzeń, od poziomu z okresu bazowego do maksimum, możliwego do uzyskania w roku obliczeniowym. Zmienne zależne podane na trzech osiach: wynagrodzenia + nagrody, suma podatków i obciążeń na PFAZ, opłacenie wypłat podatkami. Oznaczenie tych osi jest takie samo, jak znaki składające się na poszczególne krzywe wykresu. Po prawej stronie podano informacje uzupełniające dla każdego wariantu płacowego /linii wydruku/: kod obciążeń na PFAZ, wynik finansowy, rezerwa zysku i nagrody. Wartość nagród dla każdego wariantu płacowego obliczona jest w zależności od przyjętej funkcji celu. Najlepsze rozwiązania dla poszczególnych krzywych oznaczone są znakami tymi samymi, które tworzą krzywą, poza marginesami wykresu. Ilość wariantów płacowych na wykresie jest stała /w przeciwieństwie do kroku, o jaki zwiększana jest zmienna niezależna/. Każda krzywa rozciągnięta jest na całą szerokość wykresu.

Przeanalizujemy kilka typowych sytuacji na podstawie wy-

kresów sporządzonych dla rzeczywistych przedsiębiorstw. Rys. 1 przedstawia maksymalizację wynagrodzeń i nagród, tzn. ustatowanie wypłat na maksymalnym poziomie, przy założeniu dokonania z zysku do podziału obowiązkowych odpisów i zaspokojenia założonych potrzeb /fundusz rozwoju, socjalny, mieszkaniowy/. Badane przedsiębiorstwo jest w trudnej sytuacji, gdyż nawet utrzymanie wynagrodzeń na zeszłorocznym poziomie powoduje konieczność odpisów na PFAZ, a każda podwyżka pociąga za sobą wzrost tych obciążeń. Z wzrostem wynagrodzeń /w dół wykresu/ spada łączna suma wypłat /przesuwanie się wykresu wypłat oznaczonego gwiazdkami w lewo/. W obszarze C_1 spadek jest jednostajny co wskazuje na to, że zarówno wynagrodzenia jak i nagrody są obciążone na PFAZ wg najwyższych stawek w tabeli obciążeń. Wzrost wynagrodzeń o pewną stałą sumę powoduje wzrost podatków od płac i składki na ZUS o 65% tej sumy, spadek podatku dochodowego o 65% wzrostu wynagrodzeń, podatku od płac i składki na ZUS, odpowiedni spadek zysku do podziału i wzrost obciążenia PFAZ - a to co zostaje jest przeznaczone na nagrody i PFAZ od nich. Jako że przyrost wynagrodzeń między dwoma wierszami wykresu jest zawsze stały, przyrosty bądź spadki wyznaczonych wielkości są stałe wtedy, gdy nie zmienia się stawka obciążenia na PFAZ - co właśnie ma miejsce w obszarze C_1 . W obszarze C_2 tempo spadku wypłat rośnie, jako że maleje stopa obciążenia nagród na PFAZ. W obszarze C_3 nie ma obciążenia na PFAZ od nagród, spadek wypłat jest największy i jednostajny. Mogłoby się wydawać, że jeśli nie ma PFAZ-u od nagród to spadek wypłat /przy stałym wzroście wynagrodzeń/ powinien być mniejszy niż wtedy, gdy ten PFAZ trzeba płacić. Jak napisano wcześniej wzrost wynagrodzeń powoduje spadek zysku do podziału o pewną kwotę. Jeśli jest PFAZ od nagród to kwota ta obejmuje

spadek nagród i PFAZ-u od nich. Jeśli zaś jesteśmy w strefie bez-PFAZ-owej, to kwota ta sama co poprzednio obejmuje jedynie spadek nagród, który będzie więc większy niż poprzednio .

W obszarze C_4 nie starcza już zysku na nagrody, wypłaty więc rosną jeśli rosną wynagrodzenia.

Łatwo też zauważyć , że krzywe podatków/podwójne krzywyki / i opłacenia wypłat podatkami /"X"/y/ są niemal lustrzanym odbiciem krzywej wynagrodzeń co wynika z poprzednio prowadzonych rozważań. Z analizy wykresu wynika, że jeśli przyjąć że cały pozostały zysk do podziału chcemy przeznaczyć na nagrody i PFAZ od nich, nie należy podwyższać wynagrodzeń poza obszar C_1 , bo w innych obszarach suma wypłat gwałtownie i progresywnie spada, zaś podatki progresywne rosną. Najlepiej zaś - wynagrodzeń w ogóle nie podwyższać, wypłacić bardzo dużą "czternastkę", a wtedy wszyscy są zadowoleni: załoga dostaje największą wypłatę, przedsiębiorstwo płaci najmniej podatków, a wynik finansowy /czyli podstawa oceny przedsiębiorstwa/ jest największy. Kłopot polega tylko na tym, żeby załoga zrozumiała iż takie rozwiązanie jest dla niej najlepsze. Bardziej realna polityka finansowa polega na wypłacaniu nagród z zysku na poziomie maksymalnym, ale nie powodującym odpisów PFAZ - odpowiada to funkcji celu minimalizacji podatków i obciążeń.

Na rys. 2 przedstawiono wykres decyzyjny dla takiej funkcji, dla tego samego przedsiębiorstwa co na rys. 1.

Zgodnie z założeniem takiej funkcji celu, w większości wariantów płacowych zostaje jeszcze rezerwa zysku do podziału, którą można przeznaczyć na dodatkowe niż założono wcześniej zasilenie funduszy rozwoju, socjalnego i mieszkaniowego.

W obszarze F_1 nagrody z zysku są stałe, a więc stały wzrost

wynagrodzeń powoduje stały wzrost podatków i obciążeń, stąd wszystkie wykresy w tym obszarze są liniami prostymi. W obszarze F_2 zysku nie starcza na płacenie nagród w wysokości maksymalnej, nie powodującej odpisów na PFAZ, nagrody a więc i wypłaty łącznie, gwałtownie spadają. Jako że wysokość nagród nie ma tu wpływu na wysokość podatków i obciążeń, podatki rosną tak samo jak i w obszarze F_1 , a więc opłacenie wypłat podatkami zaczyna gwałtownie rosnać. W obszarze F_3 , gdy już w ogóle nie ma środków na nagrody, krzywa wynagrodzeń zaczyna znów rosnać/ jako że rosną wynagrodzenia/. Porównując rysunki 1 i 2 warto zauważyć, że obszar F_2 pokrywa się z C_3 , a F_3 z C_4 . Wnioski z tych dwóch rysunków są takie same. Najkorzystniej jest wynagrodzeń nie podwyższać, rekompensując to nagrodami z zysku a jeśli już zdecydować podwyżki, to nie więcej niż do granicy obszarów F_1 i F_2 / C_2 i $C_3 /$.

Zupełnie inaczej niż na rys. 1 wygląda wykres dla maksymalizacji wypłat z rys. 3 /inne dane/. Nie całkiem jednak, gdyż jak łatwo zauważyć rys. 1 jest niemal identyczny jak obszar C rysunku 4 - identyczne będą też dla tego obszaru komentarze i wnioski. W obszarze A wynagrodzenia nie powodują obciążeń na PFAZ, ich wzrost powoduje więc obniżkę podatków /spadek podatku dochodowego przewyższa wzrost podatku od płac i składki na ZUS/. Nagrody są bardzo wysokie, płacone jest obciążenie na PFAZ i spadek nagród powoduje tym większy spadek podatków i obciążeń. W obszarze B zaczyna się już PFAZ od wynagrodzeń, ale mała progresja obciążeń powoduje, że podatki nadal maleją. Obszar C został już opisany poprzednio. Granica obszarów B i C jest najlepszym rozwiązaniem gdyż przy najniższych podatkach i najmniejszym opłaceniu wypłat podatkami występują największe wypłaty. Może się to wydać dziwne, ale jest to z pewnym przybliżeniem

prawidłowość. Wynika ona z faktu, że suma środków na wypłaty i podatki jest stała; jeśli suma wypłat zaczyna spadać, to suma podatków rośnie.

Minimalizacja podatków dla tego przypadku /rys.4/ daje również wykres decyzyjny obszerniejszy niż na rys.2.

Obszar D zawiera rozwiązania w ogóle bez PFAZ; pokrywa się on z obszarem A z rys. 3. Jednakże w stosunku do rys.3 krzywa podatków spada łagodniej - jest to spowodowane brakiem spadku PFAZ-u od nagród /bo przecież PFAZ-u wcale tu nie ma/. W obszarze F, w którym zaczynają się obciążenia wynagrodzeń na PFAZ - krzywa podatków zaczyna rosnąć, jednakże opłacenie wypłat podatkami nadal maleje. Przedstawione wykresy są w miarę typowe, ale każde przedsiębiorstwo ma swoją specyfikę i wykresy będą zawsze się różnić.

Systemów informatycznych różnego typu wspomagających przedsiębiorstwo jest w Polsce sporo; mają one zazwyczaj jedną wspólną cechę; pracują dla średnich i niższych szczebli zarządzania.

Model PROFIN jest jedną z bardzo niewielu realizacji informatycznych, które są przydatne w pracy zarówno dyrekcji przedsiębiorstwa jak i czynników społecznych i to nie jako system informacyjny ale jako narzędzie wspomagania decyzji. System został uruchomiony w Instytucie Okręgowym Politechniki Gdańskiej, na maszynie MERA-400 w języku FORTRAN, pod systemem operacyjnym CROOK-4.

System posiada wiele udogodnień pozwalających na szybką i efektywną z nim pracę.

Szczegółowe informacje znaleźć można w instrukcji użytkownika systemów, autorstwa T. Rawińskiego.

Kto w przedsiębiorstwie może i powinien korzystać z systemu PROFIN? Widzimy w zasadzie kilku niezależnych partnerów:

dyrektor ekonomiczny /lub naczelny/, główny księgowy, przedstawiciele organizacji partyjnej, rady pracowniczej, Związków Zawodowych.

Ideał zaś - gdy wszyscy oni zasiadają przed ekranem komputera i w trakcie rzeczowej dyskusji podpartej wynikami obliczeń wspólnie ustalają zadawalający wszystkich wariant planowy.

Jeśli system można zaistalować w przedsiębiorstwie "na stałe" /jak w przypadku stoczni remontowej RADUNIA/ i terminale umieszczone są w odpowiednich komórkach funkcjonalnych - system może być użytkowany na codzień przez pracowników merytorycznych. Jeśli tak nie jest, wykorzystywanie systemu można prowadzić w formie sesji czy ekspertyz.

Rys. 2

4. MINIMALIZACJA ODPRAWDZANIACH ŚRÓDKÓW
 5. WYBARADZENIA 6. MAURUDY 7. - PODATKI 8. PFRAZ 9. - DPLALENIE WYPLAT PODATKARI

WYBIAUR. X	2,46	2,52	2,57	2,62	2,68	2,73	2,78	2,84	2,89
249837,058									
250497,2388									
251156,1188									
251916,4188									
252473,1188									
253131,6188									
253790,2188									
254448,7188									
255107,2188									
255765,6188									
256424,1188									
257082,6188									
257741,1188									
258399,6188									
259058,1188									
259716,6188									
260375,1188									
261033,6188									
261692,1188									
262350,6188									
263009,1188									
263667,6188									
264326,1188									
264984,6188									
265643,1188									
266301,6188									
266960,1188									
267618,6188									
268277,1188									
268935,6188									
269594,1188									
270252,6188									
270911,1188									
271569,6188									
272228,1188									
272886,6188									
273545,1188									
274203,6188									
274862,1188									
275520,6188									
276179,1188									
276837,6188									
277496,1188									
278154,6188									
278813,1188									
279471,6188									
280130,1188									
280788,6188									
281447,1188									
282105,6188									
282764,1188									
283422,6188									
284081,1188									
284739,6188									
285398,1188									
286056,6188									
286715,1188									
287373,6188									
288032,1188									
288690,6188									
289349,1188									
290007,6188									
290666,1188									
291324,6188									
291983,1188									
292641,6188									
293300,1188									
293958,6188									
294617,1188									
295275,6188									
295934,1188									
296592,6188									
297251,1188									
297909,6188									
298568,1188									
299226,6188									
299885,1188									
300543,6188									
301202,1188									
301860,6188									
302519,1188									
303177,6188									
303836,1188									
304494,6188									
305153,1188									
305811,6188									
306470,1188									
307128,6188									
307787,1188									
308445,6188									
309104,1188									
309762,6188									
310421,1188									
311079,6188									
311738,1188									
312396,6188									
313055,1188									
313713,6188									
314372,1188									
315030,6188									
315689,1188									
316347,6188									
317006,1188									
317664,6188									
318323,1188									
318981,6188									
319640,1188									
320298,6188									
320957,1188									
321615,6188									
322274,1188									
322932,6188									
323591,1188									
324249,6188									
324908,1188									
325566,6188									
326225,1188									
326883,6188									
327542,1188									
328200,6188									
328859,1188									
329517,6188									
330176,1188									
330834,6188									
331493,1188									
332151,6188									
332810,1188									
333468,6188									
334127,1188									
334785,6188									
335444,1188									
336102,6188									
336761,1188									
337419,6188									
338078,1188									
338736,6188									
339395,1188									
340053,6188									
340712,1188									
341370,6188									
342029,1188									
342687,6188									
343346,1188									
344004,6188									
344663,1188									
345321,6188									
345980,1188									
346638,6188									
347297,1188									
347955,6188									
348614,1188									
349272,6188									
349931,1188									
350589,6188									
351248,1188									
351906,6188									
352565,1188									
353223,6188									
353882,1188									
354540,6188									
355199,1188									
355857,6188									
356516,1188									
357174,6188									
357833,1188									
358491,6188									
359150,1188									
359808,6188									
360467,1188									
361125,6188									
361784,1188									
362442,6188									
363101,1188									
363759,6188									
364418,1188									

Rys. 3

3. BAKSTALIZACJA WYHAGRUDZEN I MAURODY
 - WYHAURODZANIA + MAURODY
 - PODAŃKI + PTAZ
 X - OPLACENIE WYPLAT PODDAWANI

MAURODY	2.19	2.23	2.27	2.31	2.35	2.39	2.43	2.46	2.50	WYH.	WYH. WYH. FIM	REZ. ZYBU	MAURODY
240397.0										1.	206354.9	0.0	63470.0
251711.1										2.	833231.8	0.0	63470.0
251712.0										3.	821431.8	0.0	63470.0
255009.7										4.	819521.7	0.0	63458.1
256302.4										5.	817417.7	0.0	63292.2
257595.0										6.	815310.6	0.0	63126.3
258007.7										7.	813203.6	0.0	62960.4
260180.4										8.	811096.5	0.0	62794.4
261473.0										9.	808989.5	0.0	62628.5
262765.7										10.	806882.4	0.0	62462.6
264058.4										11.	804775.4	0.0	62296.6
265351.0										12.	802668.3	0.0	62130.7
267946.4										13.	800561.3	0.0	61964.8
268259.0										14.	798454.2	0.0	61798.8
270524.7										15.	796347.2	0.0	61632.9
271817.4										16.	794240.1	0.0	61467.0
273110.1										17.	792133.0	0.0	61301.1
274402.8										18.	790026.0	0.0	61135.1
275695.5										19.	787918.9	0.0	60969.2
276988.2										20.	785811.9	0.0	60803.3
278280.9										21.	783704.8	0.0	60637.4
279573.6										22.	781597.8	0.0	60471.4
279895.1										23.	779490.7	0.0	60305.5
279727.7										24.	777383.7	0.0	60139.6
279570.4										25.	775276.6	0.0	59973.6
280853.1	IX									26.	773169.6	0.0	59807.6
282135.8	IX									27.	771062.5	0.0	59641.6
283418.4	IX									28.	768955.5	0.0	59475.7
283741.1	IX									29.	766848.4	0.0	59309.7
284013.8	IX									30.	764741.4	0.0	59143.8
284286.5	IX									31.	762634.3	0.0	58977.8
284559.2	IX									32.	760527.2	0.0	58811.8
284831.9	IX									33.	758420.2	0.0	58645.8
285104.6	IX									34.	756313.1	0.0	58479.8
285377.3	IX									35.	754206.1	0.0	58313.8
285650.0	IX									36.	752099.0	0.0	58147.8
285922.7	IX									37.	750000.0	0.0	57981.8
286195.4	IX									38.	747900.0	0.0	57815.7
287478.1	IX									39.	745799.9	0.0	57649.7
288760.8	IX									40.	743699.9	0.0	57483.7
289033.5	IX									41.	741599.9	0.0	57317.7
289306.2	IX									42.	739499.9	0.0	57151.7
289578.9	IX									43.	737399.9	0.0	56985.7
289851.6	IX									44.	735299.9	0.0	56819.7
290124.3	IX									45.	733199.9	0.0	56653.7
290397.0	IX									46.	731099.9	0.0	56487.7
290669.7	IX									47.	728999.9	0.0	56321.7
290942.4	IX									48.	726899.9	0.0	56155.7
291215.1	IX									49.	724799.9	0.0	55989.7
291487.8	IX									50.	722699.9	0.0	55823.7
291760.5	IX									51.	720599.9	0.0	55657.7
292033.2	IX									52.	718499.9	0.0	55491.7
292305.9	IX									53.	716399.9	0.0	55325.7
292578.6	IX									54.	714299.9	0.0	55159.7
292851.3	IX									55.	712199.9	0.0	54993.7
293124.0	IX									56.	710099.9	0.0	54827.7
293396.7	IX									57.	707999.9	0.0	54661.7
293669.4	IX									58.	705899.9	0.0	54495.7
293942.1	IX									59.	703799.9	0.0	54329.7
294214.8	IX									60.	701699.9	0.0	54163.7
294487.5	IX									61.	699599.9	0.0	53997.7
294760.2	IX									62.	697499.9	0.0	53831.7
295032.9	IX									63.	695399.9	0.0	53665.7
295305.6	IX									64.	693299.9	0.0	53499.7
295578.3	IX									65.	691199.9	0.0	53333.7
295851.0	IX									66.	689099.9	0.0	53167.7
296123.7	IX									67.	686999.9	0.0	53001.7
296396.4	IX									68.	684899.9	0.0	52835.7
296669.1	IX									69.	682799.9	0.0	52669.7
296941.8	IX									70.	680699.9	0.0	52503.7
297214.5	IX									71.	678599.9	0.0	52337.7
297487.2	IX									72.	676499.9	0.0	52171.7
297759.9	IX									73.	674399.9	0.0	52005.7
298032.6	IX									74.	672299.9	0.0	51839.7
298305.3	IX									75.	670199.9	0.0	51673.7
298578.0	IX									76.	668099.9	0.0	51507.7
298850.7	IX									77.	665999.9	0.0	51341.7
299123.4	IX									78.	663899.9	0.0	51175.7
299396.1	IX									79.	661799.9	0.0	51009.7
299668.8	IX									80.	659699.9	0.0	50843.7
299941.5	IX									81.	657599.9	0.0	50677.7
300214.2	IX									82.	655499.9	0.0	50511.7
300486.9	IX									83.	653399.9	0.0	50345.7
300759.6	IX									84.	651299.9	0.0	50179.7
301032.3	IX									85.	649199.9	0.0	50013.7
301305.0	IX									86.	647099.9	0.0	49847.7
301577.7	IX									87.	644999.9	0.0	49681.7
301850.4	IX									88.	642899.9	0.0	49515.7
302123.1	IX									89.	640799.9	0.0	49349.7
302395.8	IX									90.	638699.9	0.0	49183.7
302668.5	IX									91.	636599.9	0.0	49017.7
302941.2	IX									92.	634499.9	0.0	48851.7
303213.9	IX									93.	632399.9	0.0	48685.7
303486.6	IX									94.	630299.9	0.0	48519.7
303759.3	IX									95.	628199.9	0.0	48353.7
304032.0	IX									96.	626099.9	0.0	48187.7
304304.7	IX									97.	623999.9	0.0	48021.7
304577.4	IX									98.	621899.9	0.0	47855.7
304850.1	IX									99.	619799.9	0.0	47689.7
305122.8	IX									100.	617699.9	0.0	47523.7

4. REALIZACIJA OPRIMOVAZANÝCH SRUŽOV
 - USTANOVENIA & MAJUDY

X - OPLAČENIE VÝPLAT PODRUKANI

Rys. 4

271767.8 281039.1 289310.4 294581.6 302952.9 310121.2 317395.4 324666.7 331938.0
 652784.4 672289.2 688754.1 705238.9 721725.7 738208.5 754693.4 771178.2 787663.0

2.14 2.18 2.23 2.28 2.32 2.37 2.41 2.46 2.50

249839.0 251131.7 252424.3 253717.0 255009.7 256302.4 257595.1 258887.8 260180.5 261473.2 262765.9 264058.6 265351.3 266644.0 267936.7 269229.4 270522.1 271814.8 273107.5 274399.2 275692.0 276984.7 278277.4 279570.1 280862.8 282155.5 283448.2 284740.9 286033.6 287326.3 288619.0 289911.7 291204.4 292497.1 293789.8 295082.5 296375.2 297667.9 298960.6 300253.3 301546.0 302838.7 304131.4 305424.1 306716.8 308009.5 309302.2 310594.9 311887.6 313180.3 314473.0

MAJUDY
 249839.0 251131.7 252424.3 253717.0 255009.7 256302.4 257595.1 258887.8 260180.5 261473.2 262765.9 264058.6 265351.3 266644.0 267936.7 269229.4 270522.1 271814.8 273107.5 274399.2 275692.0 276984.7 278277.4 279570.1 280862.8 282155.5 283448.2 284740.9 286033.6 287326.3 288619.0 289911.7 291204.4 292497.1 293789.8 295082.5 296375.2 297667.9 298960.6 300253.3 301546.0 302838.7 304131.4 305424.1 306716.8 308009.5 309302.2 310594.9 311887.6 313180.3 314473.0
 MAJUDY
 249839.0 251131.7 252424.3 253717.0 255009.7 256302.4 257595.1 258887.8 260180.5 261473.2 262765.9 264058.6 265351.3 266644.0 267936.7 269229.4 270522.1 271814.8 273107.5 274399.2 275692.0 276984.7 278277.4 279570.1 280862.8 282155.5 283448.2 284740.9 286033.6 287326.3 288619.0 289911.7 291204.4 292497.1 293789.8 295082.5 296375.2 297667.9 298960.6 300253.3 301546.0 302838.7 304131.4 305424.1 306716.8 308009.5 309302.2 310594.9 311887.6 313180.3 314473.0

249839.0 251131.7 252424.3 253717.0 255009.7 256302.4 257595.1 258887.8 260180.5 261473.2 262765.9 264058.6 265351.3 266644.0 267936.7 269229.4 270522.1 271814.8 273107.5 274399.2 275692.0 276984.7 278277.4 279570.1 280862.8 282155.5 283448.2 284740.9 286033.6 287326.3 288619.0 289911.7 291204.4 292497.1 293789.8 295082.5 296375.2 297667.9 298960.6 300253.3 301546.0 302838.7 304131.4 305424.1 306716.8 308009.5 309302.2 310594.9 311887.6 313180.3 314473.0

inż. Bogdan Machowiak
Biuro Projektów Budownictwa Wiejskiego
Poznań, ul. Piekary 17
tel. 33 -05 - 81 wew.569

WARIANTOWANIE SEKCJI MIESZKANIOWYCH PRZY ZADANYCH MODUŁACH WIELKIEJ PŁYTY

W każdym procesie projektowym o charakterze typizacyjnym ważną sprawą jest możliwość uzyskania możliwie wszystkich wariantów rozwiązań projektowych. Dopiero na podstawie tych danych, po dokonaniu odpowiedniej analizy, istnieje możliwość wyboru optymalnych rozwiązań projektowych.

Właśnie taka kolejność wykonywania projektów typowych dla budownictwa wielkopłytowego implikuje powstanie bardziej funkcjonalnych budynków i sekcji a co z kolei winno prowadzić do przygotowania wciąż nowych, doskonalszych elementów składowych mieszkań, a także nowych elementów konstrukcyjnych budynków.

W Biurze Projektów Budownictwa Wiejskiego w Poznaniu opracowano na emc MERA-100 pakiet programów umożliwiający dokonanie wariantowania rysunków mieszkań "WSM" w budownictwie wielkopłytowym /nazwa programu "WSM"/.

Zasadniczą częścią pakietu programów "WSM" jest procedura sporządzania rysunków /schematów/ mieszkań - rysowanych w skali 1 : 50. Ponieważ w projektowaniu architektonicznym podstawowym środkiem wyrazu jest rysunek /schemat/, procedurze wykonywania rysunków poświęcono w "WSM" największą uwagę. Rysunki wykonywano przez program "WSM" przedstawione są schematycznie poprzez rysunek ścian zewnętrznych / w tym loggie/ i ścian wewnętrznych konstrukcyjnych bez znaczenia otworów okiennych i drzwiowych oraz zaznaczenie schematycznie "wejścia". Przy czym rysunki wykonywane są / z małym błędem/ w skali 1 : 50. Błąd skali wynika z zastosowania powszechnego urządzenia drukującego typu DZM-180/KSR/. stosowanego w standardowych zestawach MERY - 400.

Tak sporządzone rysunki umożliwiają projektantowi gruntowną analizę danego schematu przedstawionego na papierze obu - stronnie perforowanym / format A3/. Na każdym arkuszu schematu podane są dane liczbowe dotyczące nr kolejnego rysunku, nr kombinacji, układ i wymiary płyty powierzchnia całkowita mieszkania oraz wymiar 'klatki schodowej'.

Pozwala to na dokonanie wstępnych prac nad architekturą wnętrza - np. umieszczenie i wyrysowanie węzła sanitarnego ciągów komunikacyjnych, kuchni itp. Taki rysunek więc staje się źródłem który inspiruje projektanta architekta do rozwiązań optymalnych.

Projektant mając do dyspozycji wszystkie warianty mieszkań, które są możliwe do zaprojektowania z zadanych modułów wielkiej płyty oprócz tego , że dokona / po analizie/ wyboru najlepszych, optymalnych rozwiązań będzie miał pewność, że analizował wszystkie warianty mieszkań.

Taki przebieg procesu projektowego powinien prowadzić do optymalnych, lepszych, bardziej funkcjonalnych i pełnych rozwiązań architektonicznych.

W pakiecie programów "WSM" oprócz procedury wykreślenia, znajdują się bloki: konserwacyjnego wprowadzenia danych, procedura obliczeń statystycznych oraz blok generacji wszystkich wariantów.

Blok konserwacyjnego wprowadzenia danych umożliwia zadanie parametrów wielkiej płyty /wymiary/ oraz ustalenie następujących ograniczeń w wariantowaniu.

- minimalna i maksymalna powierzchnia całkowita mieszkań
- wymiary klatki schodowej.
- wyłączenie lub włączenie pewnych modułów i grup płyt / np. loggie/

Zastosowanie konwersacyjnego sposobu wprowadzania danych w "WSM" pozwala projektantowi na bezpośrednie sterowanie pracą programu łącznie z wirtualizacją "próbnych przebiegów" na monitorze ekranowym i późniejsze drukowanie odpowiednich lub wszystkich wariantów mieszkań.

Blok obliczeń statystycznych umożliwia dostarczenie projektantowi obliczeń wariantowych w postaci tabelarycznej bez sporządzenia rysunków sekcji. Projektant mając taki tabulogram posiada dokładną orientację o wyliczonych wariantach /ilości, parametry mieszkań / i może wtedy odpowiednio sterować pracą programu poprzez blok konwersacji, np. celem sporządzenia wszystkich lub części rysunków.

W bloku generacji wariantów zastosowano algorytm wyszukiwania wszystkich rysunków mieszkań według danych wejściowych i żądanych ograniczeniach. W bloku tym zastosowano metodę "systematycznego przeszukiwania",

Pakiet programów "WSM" zrealizowany został w języku programowania PORTRAN pracującym w systemie operacyjnym CROOK-4 na maszynie cyfrowej MERA-400. Opracowanie pakietów programu WSM zostało wykonane w taki sposób, aby możliwa była ewentualna implementacja na inne środowiska sprzętowe i operacyjne posiadająca kompilator języka PORTRAN.

W istniejącej postaci "WSM" umożliwia dokonanie wariantowania mieszkań w ramach następujących ograniczeń;

- do 500 rysunków mieszkań
- w zakresie do 120 /m²/ powierzchni całkowitej mieszkań
- dowolnych modułów wielkiej płyty.
- dowolnych wymiarów klatek schodowych

Ze względu na stosowane urządzenie drukujące typu DZM-180 /KSR/ czas sporządzenia przykładowych 100 rysunków wariantów mieszkań wynosi ok. 4 godzin.

Natomiast czas wykonania obliczeń statystycznych dla przykładowych 100 wariantów wynosi ok 10 minut łącznie z uzyskaniem tabulogramu.

Praca "WSM" realizowana jest w trybie konwersacyjnym umożliwiającym wprowadzenie następujących danych w "bloku wprowadzania danych"

- tytuł przebiegu,
- minimalna i maksymalna powierzchnie całkowite mieszkań
- wymiary modułów wielkiej płyty.
- wymiary klatki schodowej , na której następuje łączenie sekcji,
- wyłączenie pewnych typów modułów w żądanych ścianach,
- uwzględnienie lub odrzucenie modułów typu "loggia" w rysunkach mieszkań.

Realizacja konwersacyjna "WSM" umożliwia wybiórczo wykonywanie następujących ZLECEŃ:

- DAN - wprowadzanie danych
- OBL - obliczenia statyczne
- RYS - sporządzenie rysunków wariantów mieszkań / wybiórczo lub całościowo/
- TAR - Wydruk danych obliczeń statycznych
- RES - restart programu /np.po awarii sprzętu/

W trakcie pracy "WSM" korzysta z roboczego zbioru pamięci dyskowej o dostępie sekwencyjnym.

W Biurze Projektów Budownictwa Wiejskiego w Poznaniu na podstawie rysunków wykonanych przy pomocy polskich programów WSM wykonano katalog mieszkań dla potrzeb budownictwa mieszkaniowego na białej wielkiej płycie REM - 75/85.

mgr inż. Mirosław Biernat

dr inż. Ryszard Radomski

Instytut Chemii Organicznej i Fizycznej

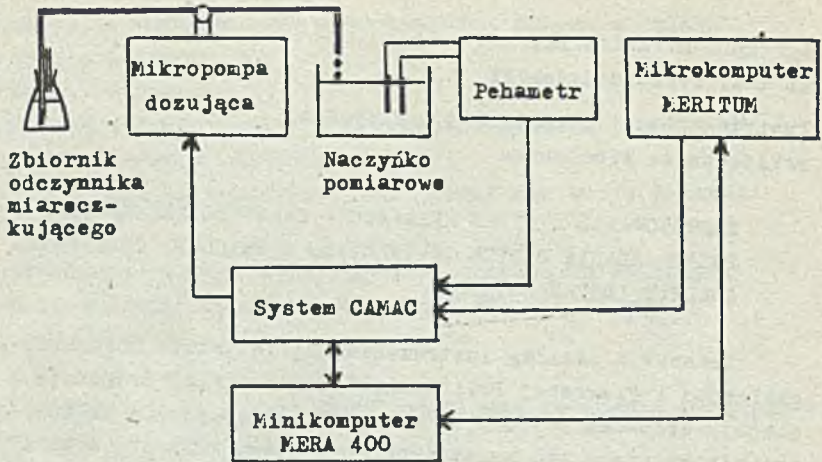
Politechnika Wrocławska

ZASTOSOWANIE SYSTEMU MERA-400 - CAMAC DO ZBIERANIA I
PRZETWARZANIA DANYCH POMIAROWYCH W PRACOWNI CHEMICZNEJ
ANALIZY INSTRUMENTALNEJ

Pracownia Analizy Instrumentalnej Instytutu Chemii Organicznej i Fizycznej Politechniki Wrocławskiej dysponuje obecnie zestawem 34 różnych metod analitycznych. Od kilku lat prowadzimy prace nad zastosowaniem systemu MERA-400 - CAMAC do skomputeryzowania wybranych stanowisk pomiarowych. W niniejszej pracy pragniemy przedstawić opracowany przez nas program przeznaczony do zbierania danych pomiarowych, sterowania eksperymentem i prezentacji wyników pomiarów. Program jest uniwersalny i może obsługiwać stanowiska pomiarowe, jakimi aktualnie dysponujemy. Należy jednak zaznaczyć, że obecnie nie mamy metod wymagających szybkiego zbierania danych, jak np. impulsowy NMR. Metody stosowane u nas są powolne pod względem szybkości rejestracji danych przez m.c. Działanie programu zostanie przedstawione na przykładzie stanowiska do miareczkowania potencjometrycznego, ponieważ jest to metoda analityczna wymagająca stosunkowo złożonej wersji tego programu.

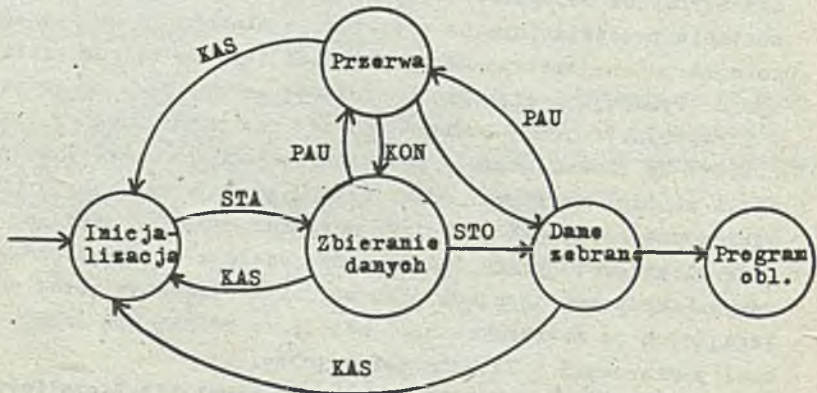
Na rys. 1 przedstawiono schemat skomputeryzowanego stanowiska do miareczkowania potencjometrycznego. W skład stanowiska wchodzi naczynko pomiarowe, do którego dozowany jest mikropompą odczynnik miareczkujący, oraz pehametr mierzący siłę elektromotoryczną ogniwa zanurzonego w naczynku pomiarowym. Mikropompa dozująca jest wspólnym elementem metod polegających na miareczkowaniu; metody te różnią się naczynkami pomiarowymi i detektorami sygnału.

Mikrokomputer MERITUM pełni rolę terminalu "inteligentnego" przy danym stanowisku pomiarowym. Całość, tj. stanowisko pomiarowe i terminal zarządzana jest przez system MERA-400 - CAMAC.



Rys.1. Schemat skomputeryzowanego stanowiska do miareczkowania potencjometrycznego

MERITUM jako urządzenie wejścia - wyjścia pozwala na wprowadzanie zleceń eksperymentatora sterujących przebiegiem pomiarów i realizujących przejścia pomiędzy stanami eksperymentu zgodnie z diagramem zamieszczonym na rys.2.



Rys.2. Diagram stanów eksperymentu pomiarowego

Po wywołaniu programu możliwe są następujące zlecenia eksperymentatora :

START - rozpoczęcie fazy zbierania danych pomiarowych,

PAUSA - zatrzymanie pomiaru na czas nieokreślony,

KONTynuacja - wznowienie pomiaru,

STOP - wcześniejsze od założonego zakończenie pomiaru z zachowaniem zebranych danych,

KASowanie - przerwanie pomiaru z przejściem do fazy wprowadzania danych.

Zlecenia te są kodowane i przesyłane do bloku 303 w systemie CAMAC z wykorzystaniem interfejsu równoległego MERITUM.

MERITUM pełni również funkcję monitora graficznego wyświetlającego na bieżąco przebieg mierzonej zależności. Mikrokomputer odczytuje dane z MERY 400 i przelicza je na dostępną w MERITUM grafikę o rozdzielczości 128x40 punktów, co w zupełności wystarcza do sprawowania kontroli nad prowadzonym eksperymentem. Skala osi Y jest przeliczana w trakcie pomiaru i po jego zakończeniu tak, aby wykres wykorzystywał całą dostępną wysokość ekranu. W trakcie trwania eksperymentu wyświetlane są informacje dotyczące realizowanego zlecenia oraz informacje o liczbie pomiarów pozostałych do zakończenia.

Wspólnymi cechami wszystkich miareczkowań jest : 1^o wskazania detektora osiągają wartość ustaloną po upływie pewnego czasu od momentu wstrzyknięcia dozy odczytanika miareczkującego, 2^o czas potrzebny na osiągnięcie stanu ustalonego zmienia się w trakcie miareczkowania; na początku analizy i po przekroczeniu punktu równoważnikowego jest krótszy, niż w okolicy punktu równoważnikowego. Z tych powodów opracowany przez nas program zawiera trzy następujące wersje :

I. Zbieranie danych pomiarowych w równoodległych odstępach czasu (ΔT) bez sterowania urządzeniami zewnętrznymi.

Ta wersja programu przeznaczona jest do rejestracji wyników pomiarów z takich przyrządów jak : spektrofotometry, chromatografy, polarografy itp. ΔT może zawierać się w granicach od 50 msek do czasu dowolnie długiego.

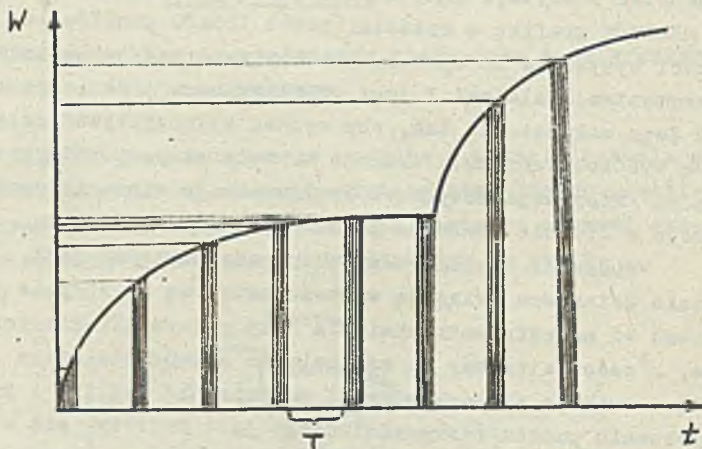
II. Zbieranie danych pomiarowych w równoodległych odstępach czasu z możliwością włączania i wyłączenia urządzenia zew-

nętrznego (np. mikrofony dozujucej, mikrokalorymetru skaningowego itp.).

Odstępy czasu w jakich następuje próbkowanie sygnału oraz uruchamianie urządzeń zewnętrznych są dobieralne. W przypadku miareczkowań ta wersja programu zasadnicze przeznaczenia jest do zgrubnych pomiarów. Można tą metodą wykonać również miareczkowanie dokładne, ale kosztem zwiększonego czasu.

III. Zbieranie danych pomiarowych z badaniem stanu ustalonego i z możliwością włączania i wyłączania urządzenia zewnętrznego. Urządzenie zewnętrzne zostaje włączone po osiągnięciu stanu ustalonego (miareczkowanie) lub po odchyleniu się od stanu ustalonego (pH-stat).

Algorytm III realizuje pomiary według rys. 3.



Rys. 3. Zmiana wskazań detektora w funkcji czasu w miareczkowaniu potencjometrycznym (rysunek schematyczny)

Komputer rejestruje 10 kolejnych wyników w odstępach czasu 200 msek (wartość ta jest limitowana przetwornikiem analogowo-cyfrowym), oblicza średnią arytmetyczną i ją zapamiętuje. Po zadany okresie T powtarza powyższą operację i porównuje wartości średnie. Jeśli dwie kolejne różnice wartości średnich są mniejsze od pewnej liczby podanej przez eksperymentatora jako dopuszczalne odchylenie, to końcowa wartość

jest zapamiętywana i uruchamiana jest pompa dozująca; jeśli różnica średnich jest większa od założonego odchylenia, to oczekiwany jest kolejny okres T i pomiar powtarza się.

Przetwarzanie danych pomiarowych oraz metody uzyskiwania z nich informacji analitycznych będą przedmiotem osobnych komunikatów (1 - 4).

Literatura

1. M. Radomska, P. Nowak, S. Janicki, R. Radomski, Zestaw programów do wygładzania i różniczkowania danych pomiarowych, Materiały "III Konferencji Użytkowników Minikomputera MERA-400", Gdańsk, 1985.
2. M. Biernat, A. Cehak, M. Radomska, R. Radomski, T. Wójcik, Skomputeryzowane miareczkowanie potencjometryczne, Komunikat zgłoszony na konferencję "Mini- i mikrokomputery w chemii", Wrocław, 1986.
3. P. Nowak, M. Radomska, Oznaczanie ziarnistości emulsji fotograficznej z zastosowaniem widna Wienera, Komunikat zgłoszony na konferencję "Mini- i mikrokomputery w chemii", Wrocław, 1986.
4. S. Janicki, M. Biernat, M. Radomska, R. Radomski, T. Wójcik, Komputerowe rozdzielanie pasm złożonych na składowe o kształcie krzywej Gaussa na przykładzie chromatogramów i elektronicznych widm absorpcyjnych, Komunikat zgłoszony na konferencję "Mini- i mikrokomputery w chemii", Wrocław, 1986.

mgr inż. Jan Kocój
dr Zbigniew Werfel
Huta Szkła Okiennego "Sandomierz"
27-610 Sandomierz
ul. Portowa 24
tel. 3041 / 143

WYKORZYSTANIE MINIKOMPUTERA MERA400 I KANAŁU INTELDIGIT PI
DO STEROWANIA W CZASIE RZECZYWISTYM
WIELOMA SPEKTROMETRAMI DLA OZNACZANIA SKŁADU CHEMICZNEGO

1. Założenia.

Niniejszy system, określany nazwą SPEKTRUM, powstał w wyniku zlecenia Centralnego Laboratorium Huty Stalowa Wola. Prace prowadzone były w ramach TNOiK przez zespół wykonawców pod kierownictwem autorów komunikatu. Podstawowe zadanie pracy sprowadzało się do:

1.1 Sprzężenie i uruchomienie współpracy

- minikomputera MERA400 w konfiguracji podstawowej /32 kbów pamięci operacyjnej, pamięć dyskowa MERA 9425, monitor DZM18OKSR, drukarka DZM180, system operacyjny SOM-3/ z pakietem jednostki sterującej kanału Inteldigit PI CA-PI3, dostarczonej przez Biuro Generalnych Dostaw CNPTKiP w Warszawie,
- kanału automatyki INTELDIGIT PI, skonfigurowanego i dostarczonego przez Zakłady Automatyki Przemysłowej MERA-ZAP w Ostrowiu Wielkopolskim,
- pracujących w Laboratorium spektrometrów ARL31000 szt.2 i ARL72000 szt.1 .

1.2 Opracowanie oprogramowania dla

- automatycznego sterowania pomiarami spektrometrycznymi,
- automatyzacji obsługi analitycznej dla oznaczania składu chemicznego próbek na podstawie wyników pomiarów,
- identyfikacji efektów międzypierwiastkowych i krzywych kalibracyjnych,

- przetwarzania statystycznego wyników analiz,
- testowania powstałego systemu .

1.3 Wdrożenie systemu i przeszkolenie obsługi.

Opracowany system czasu rzeczywistego spełnia następujące zadania:

- a/ sterowanie warunkami analizy i reżimem pracy spektrometrów,
- b/ odczyt wartości natężenia linii widmowych, mierzonych przez spektrometry i ich przeliczanie na zawartości %-owe pierwiastków w próbce z uwzględnieniem wpływu efektów międzypierwiastkowych i interferencyjnych,
- c/ automatyczna eliminacja wpływu dryftu instrumentalnego,
- d/ testowania powtarzalności pomiarów /odchylenia standardowe-ge/ i wydawanie decyzji o dodatkowych analizach,
- e/ emisja informacji o wynikach analiz,
- f/ raportowanie i archiwowanie wyników analiz,
- g/ możliwość wydruku wyników analiz na oddalonych stacjach odbierczych .

Zaprojektowany system sterowania zabezpiecza niezależność i jednoczesność funkcjonowania 3 szt. spektrometrów z utrzymaniem koncepcji rutynowej obsługi i sposobu programowania spektrometrów.

Poza systemem czasu rzeczywistego tj. w trybie wsadowym pracują dwa moduły systemu Spektrum:

- a/ prowadzenie rejestru analiz składu chemicznego prób gotowych w odpowiednie długich okresach czasu i obliczanie parametrów statystycznych dla zadanych podzbiorów tego rejestru,
- b/ identyfikacja krzywych kalibracyjnych oraz efektów międzypierwiastkowych /metoda alfa-korekcji Lachance'a-Trailla i Rasberry'ego-Heinricha/ i efektów interferencji, wykonywana metodą wielokrotnej regresji krzywoliniowej.

2. Opis systemu.

2.1 Konfiguracja minikomputera MERA400.

Minikomputer MERA400 w konfiguracji jak wyżej ze specjalizowanym systemem operacyjnym SOM-3 z ekstrakodem PROCESS i dyrektywą PI zadania komunikacji umożliwiają dowolne konfigu-

rowanie kanału PI bez konieczności ponownego generowania systemu operacyjnego.

2.2 Konfiguracja kanału INTELDIGIT PI.

- 3 szt. 16-wejściowych pakietów PI23 dla odczytu wartości woltomierzy cyfrowych spektrometrów,
- 2 szt. 8-wejściowych odbiorników sygnałów PI01 i 3 szt. PI02 z dołączonymi przerwaniami zewnętrznymi dla rejestracji stanów spektrometrów. Wyróżniono następujące sygnały: początek i koniec całkowania, start-ręczna inicjacja analizy /ARL31000/ lub podniesienia ramienia /ARL72000/, stop-ręczne przerwanie analizy, sygnały poprawności stanu spektrometrów.
- 5 szt. 8-wyjściowych nadajników sygnałów cyfrowych PO05 dla: kasowania integratorów, wyboru 5 programów analizy i reżimu pracy, stop-awaryjne przerwanie analizy przez system,
- potrzebne pakiety dopasowujące i sterowniki kaset .

2.3 Struktura systemu.

Na system sterowania spektrometrami w czasie rzeczywistym składa się 10 programów, w tym: 1 pomocniczy, 1 główny nadzorujący sterowanie i wykorzystujący pozostałe 8 jako swoje nakładki /overlaye/. Współpraca systemu z jednostką PI odbywa się przy użyciu ekstrakodu PROCESS. System ten zajmuje jedno zadanie /20k/ systemu operacyjnego SOM-3. Obszar danych w PAO zadania podzielony jest na bufora rezerwowane na wyłączne użytkowanie przez dany spektrometr. Zabezpiecza to niezależność i jednoczesność obsługi spektrometrów /"mijanie się" cykli pomiarowych/. Zbiory dyskowe o dostępie bezpośrednim są dwóch typów: tablice wykorzystywane bezpośrednio przez system czasu rzeczywistego i zbiory-rejestry analiz. Zbiory obydwu typów zajmują obszary dyskowe po ok. 10 cylindrów na dysku MERA 9425.

Na moduł przetwarzania statystycznego składa się 4 programy. Moduł wykorzystuje dyskowe zbiory-rejestry analiz. Rejestr główny, w którym zbierane są analizy dla przetwarzania zajmuje ok. 80 cylindrów /5500 analiz-odpowiada w założeniu okresowi półrocznemu/, co ma charakter opcjonalny.

Na moduł identyfikacji efektów międzypierwiastkowych składa się 6 programów. Przetwarzają one właściwie sobie zbiory-rejestry i obszar roboczy dysku, zajmujące wspólnie ok. 10 cylindrów. Zbiory pomiarów wzorców kalibracyjnych są automatycznie wypełniane przez system czasu rzeczywistego.

Oprócz tego do systemu dołączone są 3 programy testujące współpracę m.c.MERA400, jednostki PI i wejść/wyjść w spektrometrach oraz 1 program pomocniczy obsługujący zbiory-rejestry.

3. Wyniki pracy.

System został uruchomiony i pracuje w Laboratorium od czerwca br. System w sposób istotny usprawnia pracę w Laboratorium i przynosi następujące efekty:

- zwiększenie szybkości wykonywania analiz przez automatyzację obsługi spektrometrów,
- zwiększenie dokładności analiz poprzez automatyzację rekaliibracji /eliminację wpływu dryftu instrumentalnego/, badanie powtarzalności i uwzględnienie efektów międzypierwiastkowych i interferencyjnych,
- możliwość dalszego automatycznego przetwarzania wyników analiz na m.c.MERA400 .

Przewidywany jest dalszy rozwój oprogramowania w kierunku optymalizacji składów chemicznych pod kątem minimalizacji surowców deficytowych i importowanych.

Zespół wykonawców nabył doświadczenie w zakresie sprzężenia obiektów poprzez kanał przemysłowy INTELDIGIT PI z minikomputerem MERA400 i sterowania nimi w czasie rzeczywistym i jest w stanie podjęcia się podobnych prac.

Dr inż. Roman Konieczny
Mgr inż. Stanisław Krawiec
Instytut Transportu
Politechnika Śląska

MODEL RUCHU POJAZDÓW W SIECI SKRZYŻOWAŃ
- ZREALIZOWANY W JĘZYKU CSL/CROOK-4

1. Wprowadzenie

W Instytucie Transportu Politechniki Śląskiej w ramach prowadzonych prac badawczych, związanych z realizacją Programu Rządowego PR-8, wystąpiła konieczność stosunkowo szybkiego opracowania komputerowego modelu symulacyjnego - umożliwiającego ocenę efektywności różnych rodzajów sterowania sygnalizacją świetlną w dużych aglomeracjach miejskich [2].

Model ten oprogramowano w języku CSL na minikomputerze MERA-400, pracującym pod kontrolą systemu operacyjnego CROOK-4 [1]. Funkcjonalnie oprogramowanie modelu składa się z trzech modułów:

- modułu edytora budowy tablic topologicznych i innych danych wejściowych,
- modułu realizującego ruch pojazdów w sieci skrzyżowań z sygnalizacją świetlną,
- modułu oceny efektywności.

2. Struktura modelu

Strukturę modelu tworzą: klasy zbiorów, tablice topologiczne oraz bloki funkcjonalne programu symulacyjnego.

2.1. KLASY ZBIORÓW

W modelu wyróżniono pięć klas zbiorów: pojazdy, sygnalizatory, kolejki, połączenia i przybycia zewnętrzne.

Obiekty klasy „pojazdy” (CLASS TIME POJ.500(6)) mogą należeć do trzech zbiorów, określających ich lokalizację ogólną: zbioru ZSIEC (pojazdy w puli poza siecią), zbioru ZKOL (pojazdy w kolejkach) oraz zbioru ZPOL (pojazdy w połączeniach). Obiekty klasy POJ posiadają 6 atrybutów:

- atrybut 1 - określa typ pojazdu (wyróżnia się 4 typy):
1 - samochód osobowy,

- 2 - samochód ciężarowy lub autobus /krótki/,
- 3 - samochód ciężarowy z przyczepą lub autobus przegubowy,
- 4 - tramwaj ,

- atrybut 2 - określa czas przybycia pojazdu do sieci,
- atrybut 3 - określa aktualnie wybraną przez pojazd relację wyjścia z kolejki,
- atrybut 4 - określa lokalizację pojazdu /numer kolejki lub numer połączenia/,
- atrybut 5 - określa czas przybycia do kolejki,
- atrybut 6 - określa numer wlotu /numer przybycia zewnętrznego, na którym zgłosił się pojazd/.

Z klasą PCU „współpracuje” tablica stanu kolejek (KOLEJKA), w której aktualizowane są zapisy ustawienia pojazdów na poszczególnych pasach ruchu.

Obiekty klasy „sygnalizatory” (CLASS TIME CYGNAL.100(1)) posiadają jeden atrybut - stan sygnalizatora (1 - światło zielone, 0 - światło czerwone).

Obiekty klasy „kolejki” (CLASS TIME KOL.100(2)) posiadają dwa atrybuty:

- atrybut 1 - określa liczbę pojazdów w kolejce,
- atrybut 2 - /wskaznik odjazdu/ określa czy w danym cyklu sygnalizacji świetlnej nastąpił przynajmniej jeden odjazd z wybranej kolejki.

Obiekty klasy „połączenia” (CLASS TIME POL.100(1)) posiadają jeden atrybut - liczbę pojazdów znajdujących się w połączeniu, tj. na odcinku między dwoma sąsiednimi skrzyżowaniami.

Obiekty klasy „przybycia zewnętrzne” (CLASS TIME PRZyb.100) nie posiadają atrybutów.

Oprócz ww. atrybutów, obiekty wszystkich klas posiadają atrybut czasu - modyfikowany przez poszczególne bloki obsługi w programie symulacyjnym, jak i standardową procedurę upływu czasu.

W aktualnej wersji programu symulacyjnego klasy zbiorów roboczo zwymiarowano dla 500 pojazdów, 100 pasów ruchu, 100 sygnalizatorów, 100 połączeń oraz 100 przybyć zewnętrznych. Konkretny wymiary wynikają z potrzeb obliczeniowych.

2.2. TABLICE TOPOLOGICZNE

Program symulacyjny realizujący model ruchu pojazdów w sieci skrzyżowań z sygnalizacją świetlną korzysta z czterech tablic topologicznych:

- WEKOLP - tablica wejść do kolejek na przybyciu zewnętrznym,
- WEKOL - tablica wejść do kolejek na przybyciu wewnętrznym /w obrębie sieci/,
- WYREL - tablica wyboru relacji przy wychodzeniu z kolejki,
- KOLIZ - tablica kolizji.

Tablica WEKOLP o wymiarach 100 x 5 zawiera numery kolejek do których można wejść na danym przybyciu zewnętrznym /tj. wlocie do sieci/. Cztery pozycje przeznaczone są na numery kolejek dla samochodów, pozycja piąta przeznaczona jest do ewentualnego przybycia tramwaju. Podobnie zorganizowana jest tablica WEKOL o wymiarach 100 x 7, która dodatkowo na pozycji 6 zawiera długość połączenia, a na pozycji 7 maksymalną dozwoloną prędkość na tym połączeniu.

Tablica WYREL o wymiarach 200 x 8 zawiera na pozycjach 1 do 4 numery relacji wyjścia z kolejki, na pozycjach 5 i 6 efektywne czasy światła zielonego i czerwonego sygnalizatora dla danej kolejki, na pozycji 7 wskaźnik zielonej strzałki, a na pozycji 8 stan początkowy sygnalizatora w momencie rozpoczęcia symulacji.

Tablica KOLIZ o wymiarach 400 x 5 zawiera dla danej relacji na pozycji 1 numer połączenia lub wylotu do którego chce wjechać pojazd, na pozycjach 2 do 4 numery kolejek kolizyjnych przy zielonym świetle, a na pozycji 5 numer kolejki kolizyjnej przy zielonej strzałce.

Oprócz wymienionych powyżej podstawowych tablic topologicznych, program symulacyjny korzysta z tablic pomocniczych z danymi pomiarowymi zawierających m.in. zakresy czasu przybyć pojazdów różnych typów dla danego wlotu, czasy ewakuacji pojazdów z danej kolejki itp.

2.3. BLOKI FUNKCJONALNE PROGRAMU SYMULACYJNEGO

Moduł realizujący ruch pojazdów w sieci skrzyżowań składa się z następujących bloków funkcjonalnych:

- części inicjującej,
 - bloku PRZYBYCIE ZEWNĘTRZNE,
 - bloku PRZYBYCIE WENĘTRZNE,
 - bloku PRACA SYGNALIZATORÓW,
- } część dynamiczna modelu

- bloku ODJAZDÓW,
- części końcowej.

W części inicjującej wprowadzane są dane do tablic topologicznych oraz pomocniczych. Dane te zapisane są w odpowiednich zbiorach w pamięci dyskowej. W części tej następuje też /opcjonalnie/ kontrola danych wejściowych a następnie ustawienie stanu początkowego sygnalizatorów, zapełnienie zbioru ZDIEC pojazdami oraz zerowanie zbiorów ZKOL, ZPOL oraz tablicy KOLEJKA.

W bloku PRZYBYCIE ZEWNĘTRZNE losowane są według zadanego rozkładu odstępy czasu przybyć pojazdów do poszczególnych wlotów. W momencie przybycia pojazdu losowana jest pozycja z tablicy WEKOLP. Wylosowany element tej tablicy określa numer kolejki do której wpisany zostanie pojazd. Przy wpisie sygnalizowane jest ewentualne przepełnienie kolejki. Następnie odnotowywany jest numer wlotu, którym przybył pojazd; czas przybycia do sieci równy jest w tym przypadku czasowi przybycia do kolejki. Na podstawie tablicy WYREL losowana jest relacja dla pojazdu /kierunek dalszej jazdy/. W bloku tym prowadzona jest bieżąca kontrola danych wejściowych związanych z obsługą zdarzenia PRZYBYCIE ZEWNĘTRZNE.

W bloku PRZYBYCIE WEWNĘTRZNE dokonywany jest przegląd pojazdów zapisanych w zbiorze ZPOL i sprawdzany jest ich atrybut czasu. Jeżeli atrybut ten osiągnął wartość zero /pojazd dojechał już do końca wybranego połączenia/ następuje wylosowanie odpowiedniej kolejki i umieszczenie pojazdu na końcu tej kolejki. Pozostałe czynności realizowane w tym bloku są analogiczne jak w bloku PRZYBYCIE ZEWNĘTRZNE.

W bloku PRACA SYGNALIZATORÓW zmieniają się stany sygnalizatorów w zależności od przyjętego systemu sterowania.

Najbardziej złożonym blokiem programu jest blok ODJAZDÓW. W bloku tym realizowane są odjazdy pojazdów z kolejek, jeżeli istnieją do tego warunki. Usunięcie pojazdu z kolejki następuje gdy: pojazd dojechał do linii STOPU, jest zielone światło lub zielona strzałka dla danej kolejki /i pojazd odczyta „na zieloną strzałkę"/, upłynął odstęp czasu od odjazdu pojazdu poprzedniego; w przypadku lewoskrętu sprawdzane jest czy istnieje dla wszystkich kolejek kolizyjnych odpowiednia luka czasowa większa od obliczonego czasu ewakuacji pojazdu ze skrzyżowania. W przypadku „utrudnionego lewoskrętu”, gdy pojazd - najczęściej typ 3

- nie zdażył jeszcze na skutek zaistniałych warunków opuścić skrzyżowania następuje usunięcie tego /jednego/ pojazdu z kolejki w momencie zmiany cyklu świateł /podobnie jak w sytuacji rzeczywistej pojazd taki najczęściej wymusza wyjazd/. Po opuszczeniu kolejki odnotowywana jest strata czasu oraz umieszczenie pojazdu na wybranym połączeniu lub wylocie /usunięcie z sieci/. W bloku tym wyznaczany jest czas przybycia pojazdu do następnej kolejki.

Część końcowa modułu przeznaczona jest do tworzenia statystyk symulacji, wydruku histogramów itp.

3. Uwagi końcowa

Moduł realizujący ruch pojazdów w sieci skrzyżowań zrealizowany został w taki sposób, aby dla dowolnie przyjętej sieci nie było potrzeby dokonywać żadnych zmian w jego części dynamicznej. Użytkownik korzystający z programu symulacyjnego wypełnia tylko odpowiednie tablice wejściowe na podstawie topologii sieci i danych pomiarowych, a następnie otrzymuje wyniki obracowane przez moduł oceny efektywności. Moduł oceny efektywności korzysta z wyników symulacji zarejestrowanych w pamięci dyskowej, wśród których znajdują się następujące informacje:

- stan zegara,
- typ zdarzenia /przybycie zewnętrzne, wewnętrzne, odjazd/,
- rodzaj odjazdu /zwykły, utrudniony, na zieloną strzałkę/,
- typ pojazdu,
- czas przybycia do kolejki,
- czas przybycia do sieci,
- czas opuszczenia kolejki,
- czas opuszczenia sieci,
- numer wlotu,
- numer wylotu lub połączenia,
- liczba pojazdów w kolejce,
- liczba pojazdów w połączeniu.

Podstawową jednostką czasu symulacyjnego /kwantem czasu/ jest wartość 0,1 sekundy - co umożliwia dostatecznie dokładne odwzorowanie warunków rzeczywistych. Stosunek czasu symulacyjnego do rzeczywistego jest funkcją kilku czynników m.in. topologii sieci, przyjętego kwantu czasu, częstości występowania zdarzeń w sieci, itd. Przykładowo: dla wariantu sieci składającej się z 18 kolejek

stosunek czasu symulacyjnego do rzeczywistego wynosi 1/4, natomiast dla sieci zbliżającej się do 100 kolejek stosunek ten dąży do proporcji 1/1.

Wielodostępny system operacyjny CROOK-4 minikomputera MERA-400 umożliwia jednoczesną realizację kilkunastu przebiegów procesu symulacyjnego dla różnych wariantów sieci skrzyżowań. Jedynym ograniczeniem jest wielkość aktualnie dostępnej pamięci operacyjnej. Każdy moduł symulacji sieci skrzyżowań zajmuje dla wyspecyfikowanych w referacie rozmiarów zbiorów i tablic ok. 23 K słów pamięci operacyjnej, z czego na treść programu przypada ok. 10 K; treść programu wraz z tablicami może zająć maksymalnie 64 K słów pamięci operacyjnej. Łączny czas kompilacji programu: (CSL → FORTRAN → ASSEMBLER → język wewnętrzny) nie przekracza 3 minut, co jest istotne przy nanoszeniu zmian w programie.

Przyjmując, że do dyspozycji jest minikomputer MERA-400 z pamięcią operacyjną 128 K oraz z jedną stacją dysków twardej /standard/ - możliwe jest stworzenie w języku CSL/CROOK-4 modelu sieci ok. 100 skrzyżowań - co praktycznie jest wystarczające dla każdej dużej aglomeracji miejskiej w Polsce.

W celu zwiększenia efektywności oprogramowywania tego typu modeli symulacyjnych - wskazane byłoby opracowanie translatora RATCSL dla CSL /analogicznie jak RATFOR dla FORTRANu/.

Literatura

- [1] Gocałek J., Klauziński J.: Translator języka CSL-CROOK dla minikomputera MERA-400 w systemie operacyjnym CROOK-4. Instrukcja programisty - Poznań 1984 .
- [2] Konieczny R., Krawiec S., Wojciechowski B.: Symulator ruchu pojazdów w sieci skrzyżowań z symulacją świetlną dla minikomputera MERA-400. Referat na IV Konferencję Naukową Instytutu Transportu Politechniki Warszawskiej, wrzesień 1985 .

dr inż. Maria Radomska
mgr inż. Piotr Nowak
Sławomir Janicki
dr inż. Ryszard Radomski

Instytut Chemii Organicznej i Fizycznej
Politechnika Wrocławska

ZESTAW PROGRAMÓW DO WYGŁADZANIA I RÓŻNICZKOWANIA DANYCH POMIAROWYCH

Skuteczne procedury do numerycznego wygładzania i różniczkowania wyników eksperymentalnych ciągle jeszcze stanowią problem nie do końca rozwiązany. Powszechnie znane algorytmy polegające na stosowaniu wielomianów interpolacyjnych w wielu przypadkach dają wyniki niezadawalające. W Pracowni Analizy Instrumentalnej Instytutu Chemii Organicznej i Fizycznej Politechniki Wrocławskiej od kilku lat stosujemy programy wykorzystujące wygładzające własności splotu, a uzyskiwane rezultaty, według naszej opinii, godne są polecenia.

W programach przeznaczonych do wygładzania obliczany jest splot (w) zestawu danych pomiarowych (u) zmierzonych w równoodległych odstępach z pewną funkcją wagową (p). Obliczenia wykonywane są w dziedzinie funkcji - oryginałów w/g wzoru :

$$w_i = \sum_{s=-m}^m p_s u_{i+s} \quad i=m+1, \dots, N-m-1 \quad (1)$$

lub w dziedzinie transformat Fouriera tych funkcji:

$$W(f) = P(f)U(f) \quad (2)$$

gdzie w oznacza splot funkcji p i u , duże litery oznaczają transformaty Fouriera odpowiednich funkcji, a N jest liczbą danych. W przypadku stosowania wzoru (2) konieczne jest obliczenie transformaty odwrotnej funkcji $W(f)$. Jeśli funkcja wagowa p jest tak dobrana, aby miała własności filtru, to w obu metodach uzyskuje się znaczne zwiększenie stosunku sygnału do szumu.

Wygładzające własności splotu znane są od dawna, ale popularność wygładzania w/g wzoru (1) datuje się dopiero od

roku 1964, kiedy to Savitzky i Golay /1/ opublikowali stabilaryzowane wartości funkcji p ("współczynników splatających"). Tabele współczynników splatających zostały wyprowadzone przy założeniu, że do segmentu surowych danych o długości k ($k=2m+1$) dopasowany jest wielomian stopnia n ($n=2..5$), tak, aby suma kwadratów odchyłeń miała wartość minimalną. W klasycznej wersji tej metody wygładzoną wartość oblicza się w/g wzoru (1) dla punktu i , po czym obliczenia powtarza się dla punktu $i+1$. Odpowiada to przesuwaniu wielomianu o 1 punkt wzdłuż całego zestawu surowych danych, stąd opisywana metoda znana jest pod nazwą metody ruchomego wielomianu, a w przypadku $n=1$ - pod nazwą średniej bieżącej.

Tabele podane w pracy Savitzky'ego i Golay'a zawierały pewne błędy numeryczne skorygowane w pracy /2/, ponadto ograniczały się do segmentu o długości $k \leq 25$. Program do wygładzania pod nazwą WRW opracowany przez nas napisany został w/g pracy /3/, w której podano wzory do obliczania współczynników splatających nie zawierające ograniczenia $k \leq 25$. Program WRW zawiera modyfikację w stosunku do klasycznej wersji metody ruchomego wielomianu polegającą na wprowadzeniu dodatkowego parametru l ($l=1..5$). Np. $l=2$ oznacza rozpięcie wielomianu na dwukrotnie większej liczbie punktów, niż zadeklarowano, ale tylko co drugi punkt będzie pobrany do obliczenia wartości wygładzonej.

Wygładzanie w dziedzinie transformat Fouriera polega na pomnożeniu transformaty Fouriera pierwotnego sygnału przez funkcję filtrującą $P(f)$ i wykonaniu odwrotnego przekształcenia celem powrotu do pierwotnych zmiennych. W klasycznej wersji tej metody funkcja filtrująca $P(f)$ ma postać prostokątne-go okienka o jednostkowej wysokości umieszczonego w zakresie częstotliwości od 0 do częstotliwości granicznej f_0 . Równoznaczne to jest z milczącym założeniem, że w obszarze częstotliwości od 0 do f_0 nie ma składowych pochodzących od szumu, co najczęściej nie odpowiada rzeczywistości. W opracowanym przez nas programie pod nazwą FWF zastosowano funkcję filtrującą w kształcie trapezu /4/ z parametrami f_1, f_2, h_1 i h_2 dobieranymi przez użytkownika (h_1 i h_2 są wysokościami trapezu przy częstotliwościach f_1 i f_2). Do obliczania transformat Fouriera zas-

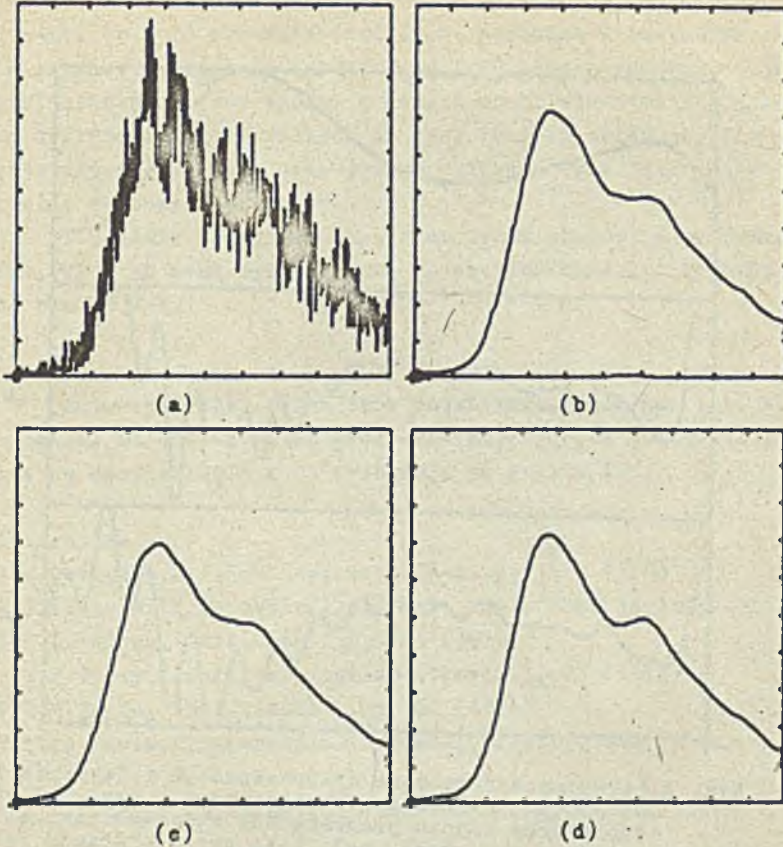
tosowano algorytm szybkiego przekształcenia Fouriera (FFT) opublikowany w pracy /5/. W algorytmach FFT zestaw danych pomiarowych traktowany jest jak okres funkcji periodycznej. Jeżeli więc zmierzony sygnał nie spełnia tego założenia, to na szłąkach rzekomych okresów powstają nieciągłości, którym w transformacie będą odpowiadać składowe o dużej częstotliwości. Opublikowane dotychczas rozwiązania tego problemu, np. w pracy /6/, wymagają dość skomplikowanych dodatkowych operacji matematycznych. W programie FWF zastosowane bardzo prosty i zarazem skuteczny zabieg polegający na tym, że jako dane wejściowe wprowadzany jest zmierzony sygnał i jego zwierciadlane odbicie. Tak spreparowane dane wejściowe nie powodują niepożądanych efektów; wymagają jedynie odrzucenia drugiej połowy danych wyjściowych.

Działanie programów WRW i FWF zademonstrowano na rys. 1. Rys. 1a przedstawia widmo fluorescencji rozdzielone w czasie kryształu mieszanego 6,5 % m akrydyny w antracenie w zakresie 400 do 510 nm zmierzony w Instytucie Fizyki PAN w Warszawie. Widmo zawiera dużo szumów, których udział można zmniejszyć kosztem wydłużonego czasu pomiaru, co jednak związane jest ze zwiększonym zużyciem ciekłego helu. Rysunki 1b, 1c i 1d przedstawiają widmo z rys. 1a wygładzone metodą ruchomego wielomianu, średniej bieżącej i filtrowania w dziedzinie transformacji Fouriera. Dane do obliczeń przedstawionych na rys. 1b, 1c i 1d wprowadzane były na taśmie papierowej; czas trwania obliczeń wynosił od kilku sekund do 1 minuty dla $N=700$.

W praktyce laboratoryjnej rzadko się zdarza, aby wygładzanie być celem końcowym; na ogół jest zabiegiem poprzedzającym różniczkowanie. W ujęciu Savitzky'ego i Gelay'a kolejne wygładzone pochodne obliczane są również jako splot w/g wzoru:

$$\frac{d^q w_i}{dx^q} = \sum_{s=-m}^m p_s^{(q)} u_{i+s} \quad i=m+1, \dots, N-m-1 \quad (3)$$

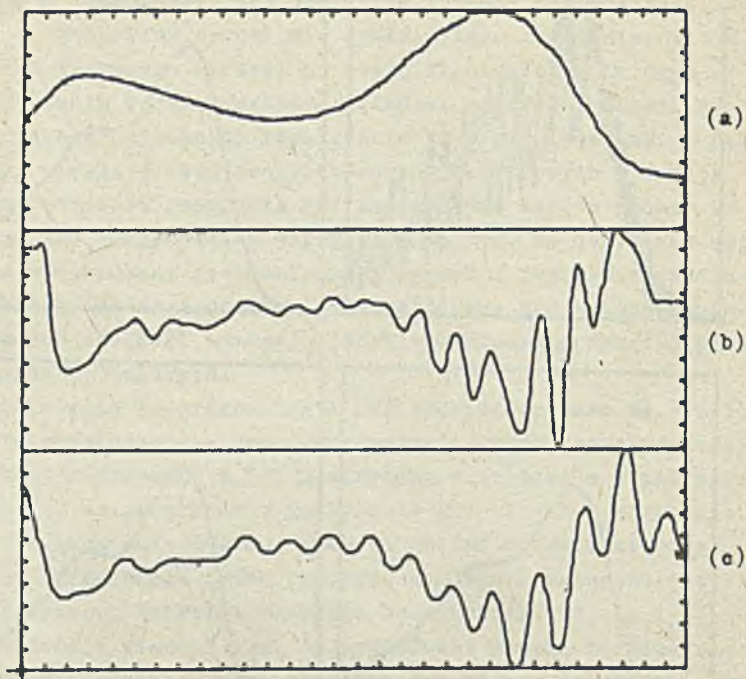
gdzie q oznacza rząd pochodnej, a $p^{(q)}$ współczynniki splatające wyznaczone z wartości współczynników wielomianów dopasowanych do segmentu surowych danych metodą najmniejszych kwadratów. Za pomocą programu WRW można obliczać kolejne wygładzone pochodne deklarując $q=1..5$ ($q=0$ oznacza operację



Rys.1. Widmo fluorescencji rozdzielone w czasie kryształu mieszanego 6,5 % m akrydyny w antracenie w zakresie 400 do 510 nm. (a) - pierwotne dane pomiarowe, (b), (c) i (d) - widmo z rys. (a) wygładzone przy użyciu programów WRW i FWF z następującymi wartościami parametrów : (b) - program WRW, dwa cykle wygładzania, w każdym $K=75$, $n=3$, $l=1$, (c) - program WRW, dwa cykle wygładzania, w pierwszym $k=25$, $n=1$, w drugim $k=13$, $n=1$, (d) - program FWF, $f_1=1$, $f_2=17$, $h_1=h_2=1$.
Objaśnienie oznaczeń - w tekście.

wygładzania). Wartości współczynników $p^{(q)}$ stosowane do obliczania pochodnych w programie WRW zaczerpnięto z pracy /3/.

Operację różniczkowania w dziedzinie transformacji można wykonać korzystając z twierdzenia o transformacji pochodnej:



Rys. 2. (a)Widmo absorpcyjne cykloheksanonu w zakresie 200 do 350 nm. (b)Druga pochodna widma z rys. (a) obliczona przy użyciu programu WRW z wartościami parametrów : $q=2$, $k=25$, $n=3$, $l=1$, przed operacją różniczkowania widmo poddano trzykrotnemu wygładzaniu stosując parametry: $k=25$, $n=3$, $l=1$. (c)Druga pochodna widma z rys. (a) obliczona przy użyciu programu RC, przed operacją różniczkowania widmo wygładzono przy użyciu programu FWF z parametrami $f_1=1$, $f_2=33$, $h_1=h_2=1$. Objasnienie oznaczeń - w tekście.

$$\mathcal{F} \left\{ \frac{d^q w(x)}{dx^q} \right\} = (j2\pi f)^q w(f) \quad (4)$$

gdzie $j = \sqrt{-1}$, symbol \mathcal{F} oznacza operację przekształcenia Fouriera. W opracowanym przez nas programie pod nazwą RC pochodne obliczane są przez wykonanie przekształcenia odwrotnego prawej strony równania (4). Dane wejściowe do programu RC muszą uprzednio zostać wygładzone metodą filtrowania w dziedzinie transformat. Na rys. 2a przedstawiono widmo cyklohek-

sanonu zarejestrowane na spektrofotometrze SPECORD UV-VIS. Rysunki 2b i 2c przedstawiają drugą pochodną widma z rys. 2a obliczoną za pomocą programów WRW i RC. Liczba składowych pasm cykloheksanonu jest zgodna z danymi opublikowanymi /7/. Dane do obliczeń przedstawionych na rys. 2b i 2c zbierane były przy użyciu programu sterującego, którego opis jest przedmiotem osobnego komunikatu.

Pośrednie etapy obliczeń oraz wynik końcowy w programach WRW, FWF i RC mogą być oglądane na oscyloskopie lub rysowane na pisaku X-Y.

Autorzy pragną serdecznie podziękować panu mgr inż. W. Cabanowi za dostarczenie programów sterujących do wyświetlenia na oscyloskopie i do rysowania na pisaku X-Y.

Literatura

1. A. Savitzky, M. J. E. Golay, *Anal. Chem.*, 36, 1627 (1964)
2. J. Steinier, Y. Termonia, J. Deltour, *Anal. Chem.*, 44, 1906 (1972)
3. H. H. Madden, *Anal. Chem.*, 50, 1383 (1978)
4. K. R. Betty, G. Horlick, *Appl. Spectrosc.*, 30, 23 (1976)
5. D. M. Monro, *Appl. Statist.*, 25, 166 (1976)
6. R. de Levie, S. Sarangapani, P. Czekał, G. Benka, *Anal. Chem.*, 50, 110 (1978)
7. H. Lischka, G. Derflinger, *Monatsh. Chem.*, 99, 2450 (1968)

Mgr Franciszek Przechowski
Dr inż. Stefan Zielinski
Instytut Okrętowy Politechniki Gdańskiej

OPROGRAMOWANIE DLA METOD OPTIMALIZACJI
STATYCZNEJ I STATYSTYKI NA MINIKOMPUTERZE
MERA-400

1. Wprowadzenie

Wiele problemów technicznych i ekonomicznych sprowadza się do zadań optymalizacji. W zadaniach takich występują trzy elementy: zbiór zmiennych decyzyjnych, których wartości mają być określone w procesie optymalizacji, kryterium jakości zwane też funkcją celu, które określa cel podejmowania decyzji oraz ograniczenia określające obszar, z którego mogą być dobierane wartości zmiennych decyzyjnych.

Matematycznie zadanie optymalizacji formuluje się następująco:

Znaleźć $\hat{x} \in X_0$ minimalizujący zadaną funkcję $f(x)$, tzn.

$$f(\hat{x}) = \min_{x \in X_0} f(x)$$

gdzie:

$x = (x_1, x_2, \dots, x_n)$ - wektor niezależnych zmiennych decyzyjnych

$f(x)$ - funkcja opisująca kryterium jakości

X_0 - obszar dopuszczalnych wartości wektora x
/ograniczenia/.

Ogólnie obszar X_0 określamy może być przez dowolny zestaw liniowych i nieliniowych równości lub nierówności.

Powstało wiele efektywnych metod numerycznego rozwiązywania zadań optymalizacji. W referacie przedstawiono jedną z nich - metodę Zangwilla.

Jedną z podstawowych metod identyfikacji i modelowania współzależnych zjawisk czy procesów jest regresja wielowymiarowa. W badaniach takich występuje dowolna ilość zjawisk pierwotnych /wejść/:

$$X_1, X_2, \dots, X_m$$

i jedno zjawisko wynikowe /wyjście/ y . Na wyjście wpływ mają też różne nieokreślone zakłócenia.

Zadanie identyfikacji /modelowania/ formuluje się następująco:

w oparciu o wyniki przeprowadzanych pomiarów:

$$X = [x_{1j}] \quad ; \quad Y = [y_j] \quad , \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, l$$

wyznaczyć takie wartości współczynników

$$a_1, a_2, \dots, a_n$$

aby model zależności zjawisk postaci:

$$\hat{y} = f(x_1, x_2, \dots, x_m, a_1, a_2, \dots, a_n)$$

najlepiej przybliżał wyniki pomiarów.

Funkcja modelu f określona jest w oparciu o analizę zjawisk lub przyjęta arbitralnie. Podstawą regresji jest tzw. zasada najmniejszych kwadratów. W referacie przedstawiono metodę obliczeniową regresji wielowymiarowej jako zadania minimalizacji, dla dowolnej postaci modelu.

W referacie omówiono również zestaw programów statystycznych opracowanych na podstawie oprogramowania firmy WANG.

Metody te mogą być stosowane przez techników i ekonomistów również do rozwiązywania problemów identyfikacji i modelowania. Wśród tych programów znajdują się programy dotyczące regresji, korelacji, analizy wariancji, kowariancji oraz programy dotyczące różnych rozkładów i testów.

2. Metoda Zangwilla

Metoda ta jest jedną z efektywniejszych zdeterminowanych iteracyjnych metod bezgradientowych szukania minimum funkcji wielu zmiennych bez ograniczeń.

Istota metody polega na kolejnym poszukiwaniu ekstremów funkcji wzdłuż kierunków wyznaczonych przez zmieniający się, bazowy układ kierunków w przestrzeni zmiennych niezależnych. Zmiany układu kierunków uczula metodę na kształt funkcji. Po każdorazowym obiegu kierunków układu bazowego, poszukuje się ekstremów wzdłuż kierunków stałego układu i sprawdza się warunki zakończenia.

Omawiany algorytm opracowano w oparciu o opis metody podany w [1 - 3] i dodatkowe jej badania przedstawione w [4]. W algorytmie wyróżnić można 4 etapy.

Etap 1. Ustalenie wartości początkowych. Utworzone zostają dwa układy ortogonalnych wektorów. Współrzędne wektorów tych układów pamiętane są w tablicach:

Q - dla zmienionego układu wektorów,

C - dla układu stałego.

Na początku układy są identyczne i pokrywają się z wersorami układu współrzędnych. Wyzerowane zostają zmienne pomocnicze:

R - służąca do sterowania zakończeniem procedury,
L - licznik iteracji. Jedna iteracja obejmuje etapy 2-4.
Wyliczana zostaje wartość funkcji w zadany przez parametr X punkcie początkowym minimalizacji.

Etap 2. Szukanie minimów wzdłuż kierunków układu stałego
i określenie dokładności

Licznik iteracji L zwiększony zostaje o 1. Współrzędne aktualnego argumentu funkcji X zapamiętywane są w tablicy X1, a aktualna wartość funkcji Y pod zmienną Y1.

Przeprowadzona zostaje minimalizacja funkcji wzdłuż kierunków stałego układu wektorów. Szukanie minimum wzdłuż dowolnego kierunku realizowane jest przez stały fragment programu /procedurę wewnętrzną/.

Następuje zmiana współrzędnych punktu zawartych w tablicy X oraz określona zostaje nowa wartość funkcji Y.

Szukanie minimum na kierunku rozpoczyna się od przesunięcia w tym kierunku o wielkość H_0 . Jeśli wartość funkcji w nowym punkcie jest mniejsza, wielkość kroku zostaje podwojona,

Czynności te powtarzane są tak długo, aż wartość funkcji po przesunięciu nie będzie mniejsza od wartości w punkcie poprzednim. Współrzędne wierzchołka paraboli wyznaczonej w oparciu o trzy ostatnie wartości funkcji /punkt XC/ zostają porównane z wartością funkcji w punkcie środkowym X. Punkt, w którym wartość funkcji jest mniejsza uznany zostaje za minimum funkcji na kierunku.

Jeżeli pierwszy krok na kierunku był nieudany, następuje przesunięcie w przeciwnym kierunku o wielkość podwojonego kroku H_0 .

Po minimalizacji wzdłuż wszystkich kierunków stałego układu wektorów obliczona jest wartość wskaźnika dokładności

$$Y_1 = \frac{|y_1 - y|}{\xi(x_1, x)}$$

gdzie $\xi(x_1, x)$ jest odległością punktów x_1 i x .

Etap 3. Sprawdzenie warunku dokładności i wydruku wyników pośrednich i końcowych

Jeśli $Y_1 < E$, gdzie E - zadana dokładność, podstawia się $R = 1$ co powoduje wydruk wyników końcowych.

Jeśli zaś $R \neq 1$ i

$$L - \text{entier}(L/L_1) \cdot L_1 < L_2$$

następuje wydruk wyników pośrednich.

Wydruki wyników wykonuje procedura PISZ, która również sprawdza ograniczenie na ilość iteracji.

Etap 4. Szukanie minimów wzdłuż kierunków układu zmiennego i zmiana tych kierunków

Jest to końcowy etap każdej iteracji. Najpierw poszukuje się minimów wzdłuż wszystkich kierunków układu zmiennego. Następnie zmienia się $N-1$ z tych kierunków wg zależności:

$$Q_j, i-1 = Q_{ji} ; \quad j = 1, 2, \dots, N, \quad i = 2, 3, \dots, N$$

a ostatni kierunek zależności

$$Q_{iN} = \frac{x_i - x_{1i}}{\xi(x_1, x)} ; \quad i = 1, 2, \dots, N$$

Zmniejszony zostaje krok H , poszukuje się minimum wzdłuż ostatniego kierunku Q_N i przechodzi do nowej iteracji czyli do etapu 2.

3. Oprogramowanie metody Zangwilla

Metoda Zangwilla oprogramowana została w języku FORTRAN-CROOK4. Programy opracowane zostały w dwóch wersjach: rozszerzonej tzw. badawczej i skróconej - roboczej.

Wersja badawcza umożliwia wyprowadzenie informacji w trakcie obliczeń. Śledzenie takie jest użyteczne przy skomplikowanych zadaniach optymalizacji, w których proces zbieżności jest bardzo powolny.

Oprogramowanie składa się z następujących segmentów:

- Procedura metody Zangwilla postaci

SUBROUTINE Z GWB (N,X,Y, IT,E,H,L1,L2)

gdzie: N - wymiarowość minimalizowanej funkcji,

X - tablica, na wejściu do procedury zawiera współrzędne punktu początkowego, na wyjściu współrzędne znalezionej punktu minimalnego,

Y - wartość funkcji w punkcie minimalnym,

IT - ograniczenie na ilość iteracji,

E - zadana dokładność określania minimum,

H - krok początkowy,

L1,L2 - parametry sterujące wyprowadzeniem wyników pośrednich: co L1 iteracji wyprowadzanych jest L2 kolejnych iteracji.

Procedura wykorzystuje segmenty zewnętrzne F, GR, PISZ.

- Procedura wydruku PISZ postaci

SUBROUTINE PISZ (N,H,E,IT,L1,L2,L,K,IR,X,F,G)

Procedura służy do wydruku danych wejściowych, wyników pośrednich i wyników końcowych.

- Procedura funkcyjna typu rzeczywistego GR postaci

FUNCTION GR (F,N,X)

Procedura określa moduł gradientu N-wymiarowej funkcji F w punkcie X.

- Procedura funkcyjna typu rzeczywistego F (X) określająca sposób wyliczania wartości minimalizowanej funkcji w punkcie X.

- Segment główny programu MASTER PZGWB

Postać robocza nie daje możliwości wyprowadzania wyników pośrednich. W związku z tym procedura minimalizacji ma mniejszą ilość parametrów i ma postać

SUBROUTINE ZGWR (N,X,Y,E,H)

Parametry N,X,Y,E odpowiadają ściśle takim samym parametrom procedury w postaci badawczej. Parametr H spełnia dodatkową funkcję, zapobiegając zapętleniu.

W przypadku, gdy ilość iteracji przekroczy wartość $100 \cdot (N+3)$ podstawia się $H = 0$, co powoduje przerwanie obliczeń.

4. Regresja wielowymiarowa

Według zasady najmniejszych kwadratów optymalne współczynniki a_1, a_2, \dots, a_n /patrz wprowadzenie/ można wyznaczyć z następującego zadania minimalizacji

$$\min_{a_1, a_2, \dots, a_n} \left\{ \sum_{i=1}^1 [Y_i - f(X_{1_i}, X_{2_i}, \dots, X_{m_i}, a_1, a_2, \dots, a_n)]^2 \right\}$$

Jak widać kryterium jakości ma tu postać sumy kwadratów odchyleń pomiędzy wartościami wyjść uzyskanych w pomiarach i odpowiednimi wartościami modelowymi określonymi dla wartości wejść

uzyskanych w pomiarach.

W przypadku modeli liniowych względem współczynników a_1, a_2, \dots, a_n powyższe zadanie minimalizacji sprowadza się do rozwiązania układu n równań algebraicznych liniowych.

Omawiany program obliczeniowy rozwiązuje powyższe zadanie minimalizacji dla dowolnej postaci modelu. Do rozwiązania wykorzystuje procedurę Zangwilla w postaci roboczej. Program napisano w języku FORTRAN-CROOK4. W skład oprogramowania wchodzi:

- segment główny /MASTER/, w którym następuje wczytanie danych wejściowych, wydruk kontrolny tych danych, wywołanie procedury Zangwilla i wydruk wyników obliczeń
- procedura funkcyjna F służąca do wyliczania wartości funkcji minimalizowanej /sumy kwadratów odchyleni/
- procedura funkcyjna FM określająca postać modelu regresji.

Procedurę przygotowuje dla swojego problemu użytkownik.

- procedura Zangwilla w postaci roboczej ZGWR.

Dane wejściowe muszą zawierać:

- dowolny tekst umieszczony w jednej linii; służy on do opisu przyjętego modelu regresji i będzie umieszczony w nagłówku wyników,

M - wymiarowość wejścia,

N - ilość współczynników,

L - ilość pomiarów,

$X_{11}, X_{21}, \dots, X_{M1}$; Y_1 - komplety danych pomiarowych / $i = 1, 2, \dots, L$ /

H - krok dla procedury minimalizacji,

E - dokładność minimalizacji,

$a_1^0, a_2^0, \dots, a_N^0$ - początkowe wartości współczynników regresji,

I - parametr sterujący; jeśli $I = 1$ nastąpi przejście do obliczeń następnego przykładu /dla tego samego modelu obiektu/ i

wczytanie nowych danych wejściowych, jeśli I ≠ 1 zakończenie obliczeń.

Jak już wspomniano wyżej w skład wyników obliczeń wchodzi wydruk kontrolny danych wejściowych /włącznie z tekstem/.

Właściwe wyniki obliczeń obejmują:

- współczynniki regresji /optymalne/,
- porównanie wartości pomiarowych y z modelami,
- współczynnik korelacji,
- wynikową sumę kwadratów /minimalną/.

Jak wskazano wyżej, optymalne wartości współczynników regresji uzyskuje się na drodze minimalizacji. Jeśli minimalizacja zostanie zakończona w wyniku przekroczenia ograniczenia na ilość iteracji 100 . /N+3/, na końcu wyników obliczeń zostanie zamieszczona informacja, iż nie uzyskano założonej dokładności E.

5. Programy statystyczne

Dla potrzeb badawczych i statystycznych w Instytucie Okrętowym opracowano zbiór programów obliczeniowych statystyki. Wykorzystano oprogramowanie firmy WANG z tego zakresu.

Programy napisano w języku BASIC. Dla uprzyęstwienia użytkownikowi obsługi programów posłużono się niezbędną konwersacją. Informacje, jakie dane i w jakiej kolejności wprowadzać pojawiają się na monitorze po uruchomieniu programów. W większości programów dane mogą być zapisane wcześniej na zbiór lub po uruchomieniu programu wprowadzane z monitora. W drugim wypadku są automatycznie również zapisywane na zbiór. Takie podejście umożliwia zmianę tylko niektórych danych /przy użyciu EDITA/ bez konieczności wprowadzania pozostałych w kolej-

nych uruchomieniach programu.

W zbiorze są następujące programy:

Program MSTA01 dotyczący regresji liniowej. Określa metodę najmniejszych kwadratów współczynniki A i B równania

$$Y = A + B X$$

na podstawie zbioru obserwacji: $Y_1, X_1, i = 1, 2, \dots, n$. Oblicza sumę kwadratu błędu, współczynniki zbliżności i korelacji oraz oszacowanie błędu standardowego. Dla zadanych X oblicza wartość Y według znalezionej modelu.

Program MSTA02 dotyczący liniowej regresji wielokrotnej. Określa metodą najmniejszych kwadratów współczynniki $B_i, i = 0, 1, 2, \dots, m$ równania

$$Y = B_0 + B_1 X_1 + \dots + B_m X_m$$

na podstawie zbioru obserwacji: $Y_1, X_{11}, X_{21}, \dots, X_{m1}, i=1, 2, \dots, n$. Pozostałe obliczenia jak wyżej.

Program MSTA03 dotyczący regresji n-tego rzędu. Określa metodą najmniejszych kwadratów współczynniki $B_i, i = 0, 2, \dots, m$ równania

$$Y = B_0 + B_1 X + B_2 X^2 + \dots + B_m X^m$$

na podstawie zbioru obserwacji $Y_1, X_1; i = 1, 2, \dots, n / n \neq m + 1$. Pozostałe obliczenia jak wyżej.

Program MSTA04 dotyczący regresji wykładniczej. Określa współczynniki A i B równania

$$Y = A \cdot e^{BX}$$

na podstawie zbioru obserwacji $Y_1, X_1; i = 1, 2, \dots, n$. Pozostałe obliczeniach jak wyżej.

Program MSTA05 dotyczący regresji geometrycznej. Określa współczynniki A i B równania

$$Y = A \cdot X^B$$

na podstawie zbioru obserwacji Y_i, X_i ; $i = 1, 2, \dots, n$. Pozostałe obliczenia jak wyżej.

Program MSTA06 dotyczący korelacji liniowej. Oblicza współczynnik korelacji liniowej dwóch zmiennych Y i X na podstawie zbioru obserwacji Y_i, X_i ; $i = 1, 2, \dots, n$.

Program MSTA07 dotyczący macierzy korelacji. Oblicza współczynniki korelacji liniowej między każdą parą z "m" zmiennych na podstawie "n" obserwacji tych zmiennych.

Program MSTA08 dotyczący modelu I analizy wariancji. Oblicza dla k próbek /grup/ sumy kwadratów i średnie kwadratowe między grupami, w grupach i całkowite oraz wartość testu F - Snedecona.

Program MSTA09 dotyczący modelu II analizy wariancji. Oblicza sumy kwadratów i średnie kwadratowe między wierszami, między kolumnami, resztowe i całkowite oraz wartość testu F dla kolumn i wierszy. Dane wejściowe podaje się w tablicy /k.n/.

Program MSTA10 dotyczący analizy wariancji według kwadratu łączącego. Oblicza sumy kwadratów i średnie kwadratowe dla wierszy, kolumn, oddziaływań wzajemnych, resztowe i całkowite oraz wartość statystyki F dla wierszy, kolumn i oddziaływań. Dane wejściowe w postaci macierzy danych i macierzy oddziaływań.

Program MSTA11 dotyczący rozkładu Chi-kwadrat. Oblicza wartość χ^2 dla rozkładu zaobserwowanego i oczekiwanego oraz prawdopodobieństwo wystąpienia danego χ^2 przy różnych stopniach swobody.

Program MSTA12 dotyczący analizy Chi-kwadrat. Oblicza wartość Chi-kwadrat dla tablicy zależności o wymiarach N.M oraz wartości oczekiwane i wpływ każdego z elementów tablicy na wielkość tego χ^2 .

Program MSTA13 dotyczący testu T-Studenta. Oblicza wartość statystyki "T" dla sprawdzenia następujących hipotez dotyczących dwóch próbek:

a/ $\mu_1 = K$, K-dana wartość

b/ $\mu_1 = \mu_2$, $\sigma_1 = \sigma_2$

c/ $\mu_1 = \mu_2$, $\sigma_1 \neq \sigma_2$

d/ $\mu_1 = \mu_2$, dla N = M

gdzie: N, M - licznosc próbek,

μ_1, μ_2 - srednie próbek,

σ_1, σ_2 - odchylenia standardowe próbek

Program MSTA14 dotyczący testu znaków i par Wilcoxon'a. Oblicza wartość testu dla n par obserwacji.

Program MSTA15 porównuje dwie próbki według testu Mann-Whitney'a.

Program MSTA16 oblicza częstość i wartość funkcji rozkładu normalnego na podstawie wartości zmiennej standaryzowanej lub średniej i odchylenia standardowego.

Program MSTA17 dotyczy ujemnego rozkładu dwumianowego. Określa prawdopodobieństwo

$$P / K, R, P / = \frac{/R + K - 1/ !}{K ! /R - 1/ !} P^R \cdot Q^k$$

gdzie: K - oczekiwana ilość zdarzeń nieudanych,

R - oczekiwana ilość zdarzeń udanych,

P - zadany podział,

$$Q = 1 - P$$

Program MSTA18 dotyczy rozkładu dwumianowego. Określa prawdopodobieństwo

$$P / K, N, P / = \frac{N!}{K! / N - K!} \cdot P^K \cdot Q^{N-K}$$

gdzie: N - ilość prób Bernoulliego,

K - ilość zdarzeń udanych,

P - zadany podział.

Program MSTA19 dotyczy rozkładu Poisson'a. Określa prawdopodobieństwo

$$P / K, \lambda / = \frac{e^{-\lambda} \cdot \lambda^K}{K!}$$

gdzie: K - częstość,

λ - oczekiwana częstość.

Program MSTA20 oblicza wartość statystyki "F" Snedecona dla N stopni swobody licznika, D stopni swobody mianownika.

Program MSTA21 oblicza wartość statystyki "T" przy N stopniach swobody dla testu dwustronnego.

Program MSTA22 generuje liczby pseudolosowe o rozkładzie normalnym ze średnią m i odchyleniem standardowym σ .

Program MSTA23 oblicza dla nieuporządkowanych danych: średnią, wariancję i odchylenie standardowe.

Program MSTA24 oblicza dla pogrupowanych /powtarzających się/ danych: średnią, wariancję i odchylenie standardowe.

Program MSTA25 oblicza dla zadanej próbki obserwacji średnią geometryczną i odchylenie standardowe geometryczne, według wzorów:

$$\bar{x} = \sqrt[n]{A_1 \cdot A_2 \cdot \dots \cdot A_n}$$

$$s = e^{\sqrt{\frac{B}{n(n-1)} - \frac{(\log(\bar{x}))^2}{n(n-1)}}}$$

gdzie: A_i - wielkość w próbce

n - liczba obserwacji

$$B = \sum_{i=1}^n (\log(A_i))^2$$

Program MSTA26 określa kowariancje krzyżowe między szeregami A i B dla przesunięć $K = 0, 1, \dots, L-1$, gdzie $L < N$, N - liczba elementów w szeregach.

Program MSTA27 oblicza autokowariancję szeregu czasowego.

Przewiduje się oprogramowanie całego zestawu również w FORTRANIE-CROOK4.

LITERATURA

1. FINDEISEN W., SZYMANOWSKI J., WIERZBICKI A.: Teoria i metody obliczeniowe optymalizacji PWN, Warszawa 1977.
2. POWEL M.J.D.: An efficient method for finding the minimum of a function of several variables without calculating derivatives. The Computer Journal, nr 7, 1964.

3. ZANGWILL W.I.: Minimizing a function without calculating derivatives. The Computer Journal, nr 10, 1967.
4. JAGODA J., ZIELIŃSKI S.: Badanie algorytmów optymalizacji na uproszczonym modelu wstępnego projektowania statku. Budownictwo Okrętowe, Nr 1, 1971.
5. ZIELIŃSKI S.: Algorytmy i programy. Zeszyt 1. Statyczna optymalizacja wielowymiarowa bez ograniczeń. Prace Badawcze IO PG, Nr 1160/MR-294/79.
6. ZIELIŃSKI S.: Algorytmy i programy. Zeszyt 3. Regresja wielowymiarowa. Prace Badawcze IO PG, Nr 1297/MR-394/80.
7. JAGODA J., KOWALCZYK J.: Pakiet standardowych programów statystycznych i inżynierskich. firmy WANG. Opracowanie CTO Nr RP-79/U-029, 1979.

V KOMUNIKATY I OFERTY

M. Guja, K. Jojczyk

Katedra Informatyki

Uniwersytet Jasielloński

Kraków, Kopernika 27

MERITUM I TERMINALEM V24

Od dłuższego już czasu rozbudowa wielodostępnych systemów komputerowych jest znacznie utrudniona poprzez brak na rynku komputerowym zwykłych terminali - monitorów ekranowych z klawiaturami. Zwykle wymagane jest, aby terminale te były połączone z komputerem zgodnie z osólnie znanym standardem V24 (RS232C). Problem ten rozwiązywany był ostatnio przez wykorzystanie jako terminali 8-bitowych mikrokomputerów takich jak: PSPD-90, czy MK-45, których koszt przewyższa jednak znacznie cenę terminala.

W roku 1985 na liście mikrokomputerów oferowanych na rynku krajowym pojawił się produkt Meritum 1, tani, 8-bitowy mikrokomputer, którego jedyną zaletą jest wygodna klawiatura. Założenia konstrukcyjne czynią z niego narzędzie nieprzydatne zarówno dla profesjonalistów, ani dla amatorów.

W Laboratorium Systemów Cyfrowych Katedry Informatyki UJ zaadaptowano Meritum 1 jako terminal dowolnego komputera wyposażonego w łącze V24. Dostępne są dwie wersje terminala:

- a. wersja, w której łącze obsługiwane jest przez ładowany z kasety magnetofonowej program

b. wersja z przeprogramowaną pamięcią EPROM.

Wersja a. ma te zalety, że nadal może służyć jako Meritum 1, ale program obsługi musi być każdorazowo ładowany. Wersja b. jest już tylko terminalem.

W obu przypadkach tzw. znaki semigraficzne mogą być zastąpione małymi literami.

Po załadowaniu programu obsługi łączy (w przypadku a.) lub po włączeniu do sieci (w przypadku b.), a także w dowolnym momencie pracy terminala można zmienić parametry transmisji:

1. prędkość (do 19200 bodów)
2. długość znaku
3. rodzaj parzystości
4. liczba bitów stopu.

Dostępny jest pełny kod ASCII - 128 znaków, a w celu uczynienia pracy bardziej wygodną istnieje możliwość redefiniowania znaków klawiatury.

Obsługiwane są wszystkie funkcje ekranowe, analogiczne jak w monitorach typu MERA 7953 vad, takie jak na przykład adresowana pozycja kursora (standard VT52).

Podsumowując należy stwierdzić, że zaadaptowany mikrokomputer Meritum 1 może służyć jako tani, pełnowartościowy 64 kolumnowy terminal pracujący w standardzie V24.

mgr Leszek Czerwosz
Centrum Medycyny Doświadczalnej i Klinicznej PAN
Warszawa 00-784 Dworkowa 3

ZESTAW MAKRODEFINICJI ASSEMBLERA DO WSPÓŁPRACY
MERA - 400 - CAMAC W SYSTEMIE OPERACYJNYM SOM-3

W trakcie kompletowania systemu minikomputerowego Mera-400 i systemu pomiarowego Camac natrafiłem na poważną przeszkodę w postaci braku sprawnego oprogramowania. Do dyspozycji był jedynie SOM-3 ze specjalnymi ekstrakodami. Trudności w uzyskaniu bezbłędnie działającego "softwaru" i perspektywa ograniczeń czasowych przy realizacji programów komunikujących się z urządzeniami Camac zmusiła mnie do napisania serii makrodefinicji assemblera wykorzystujących bezpośrednio rozkazy maszyny IN oraz OU skierowane do kanału pamięciowego. Okazało się to legalne i około 50 razy szybsze w działaniu od realizacji opartej na ekstrakodach.

Zestaw makrodefinicji jest prostą ale silną obudową rozkazów IN i OU. Wykorzystuje kompilację warunkową co uelastycznia mechanizm przekazywania parametrów oraz pseudoinstrukcję DFF umożliwiającą podział słowa komputera na pola bitowe dla kompletowania w czasie kompilacji zawartości słowa przesyłanego do Camaca jako rozkaz NAF.

Z góry zrezygnowałem z transmisji blokowych jak również z obsługi przerwań. Nie można było tego zrobić aż tak prosto. Na obecnym etapie całkowicie wystarcza nam odpowiedź EN sygnalizująca brak gotowości bloku Camac lub Q=0.

Podstawowe makrodefinicje to:

CAC,CAZ,CAI - odpowiednio generuj sygnał C,Z,ustaw I na magistrali;

ZB - zeruj rejestr B sterowni a Camac;

FB /FBB/ prześlij 16 bitów/24 bity/ rejestru B do wybranego rejestru procesora maszyny;

SB /SBB,SBO/ - odwrotnie, prześlij z maszyny do B sterownika;

NAF - prześlij rozka Camac do rejestru NAF sterownika i wykonaj na magistrali.

Dodatkowo makrodefinicje INICAM i HELPER ułatwiają obsługę błędów transmisji i niesprawności urządzeń.

Pełny tekst makrodefinicji wraz z opisem działania udostępnię.

Janusz Goczałek
Jacek Klauziński
Ryszard Zenker

SYSTEM ANCO

System ANCO (ANaliza Centralnego Ogrzewania) jest przeznaczony do obliczeń inżynierskich i doboru urządzeń przy projektowaniu wodnych dwururowych instalacji centralnego ogrzewania. Wzorowany na istniejącym systemie ANKO dla emc Odra 1204.

System posiada możliwości liczenia charakterystyk cieplnych przesród, strat ciepła, listowanie danych z równoczesną sygnalizacją błędów, wykonuje zestawienia materiałowego obejmującego: długości przewodów, izolacje cieplne, liczbę zaworów itd.

Niezależnie od stosowanej techniki obliczeniowej przy projektowaniu instalacji CO, kompletuje się w pewną funkcjonalną całość zbiór różnych elementów opisanych w katalogach (np. przejeźniki, kształtki, zawory itd.). Projekt stanowi informację o sposobie skompletowania urządzenia z jednoczesnym uzasadnieniem obliczeniowym dokonanego wyboru. System ANCO obejmuje zatem język katalogów oraz język danych opisujących projekt.

Sterowanie systemem odbywa się przy pomocy opcji, które są podawane w czasie uruchamiania systemu. Każda opcja określa wykonanie odpowiedniej funkcji w systemie. Funkcje systemu to przykładowo: sprawdzenie danych, liczenie strat ciepła, drukowanie nazw katalogów, stronicowanie wydruków itp.

System posiada bogatą sygnalizację błędów. Sygnalizowane są błędy oczywiste uniemożliwiające uzyskanie wyników obliczeń oraz przypadki podania w danych wielkości, które zazwyczaj w instalacjach nie występują.

W czasie liczenia programem można otrzymać następujące wyniki:

- informacje o przebiegu liczenia
(monitorowanie przebiegu liczenia)
- obliczenia katalogu przesród
- obliczenia strat ciepła
- listę alarmów
- obliczenia hydrauliczne
- obliczenia przejeźników
- zestawienie końcowe

Jeżeli są to tylko ogólne uwagi na temat Systemu ANCO. Szczegółowe informacje użytkownik uzyska od autorów w formie opisu systemu.

Grzegorz Banowicz

**Adaptacja programu PROBUS
w systemie operacyjnym CROOK-4.**

Program analizy statycznej ustrojów sprężystych, PROBUS jest przeznaczony do prowadzenia obliczeń statycznych szerokiej klasy konstrukcji prętowych i ram, rusztów i kratownic na podłożu sztywnym i sprężystym. W ciągu wielu lat eksploatacji w przedsiębiorstwach, biurach projektów i ośrodkach obliczeniowych program ten zdobył dużą popularność wśród użytkowników. Kierując się powyższymi przesłankami, aby umożliwić eksploatację programu na minikomputerze MERA-400 we współpracy z systemem operacyjnym CROOK-4, uruchomiono program RURAK (Ruszt Rama Kratownica) będący adaptacją programu PROBUS. Dla wygody użytkowników pozostawiono niezmienną strukturę zbioru danych i tylko nieznaczne zmiany wprowadzono w postaci wydruku wyników. Zasadnicze zmiany poczyniono w wewnętrznej budowie programu, aby umożliwić pracę programu w systemie operacyjnym CROOK-4. Zmiany te, istotne dla pracy samego programu, nie są w żadnym stopniu odczuwane przez użytkownika programu. Program RURAK wykonuje wszystkie zadania, które można było wykonać przy pomocy programu PROBUS, tak więc jest narzędziem, którym mogą posługiwać się bez żadnych trudności wszyscy dotychczasowi użytkownicy poprzednich wersji PROBUS. Ponadto, uruchomienie programu RURAK jest znacznie prostsze niż dotychczasowej wersji programu PROBUS, co wynika z wprowadzonych modyfikacji, jak i z organizacji pracy w systemie operacyjnym CROOK-4.

W przyszłości przewiduje się możliwość opracowania dodatkowych procedur, umożliwiających wprowadzanie danych w sposób konwersyjny.

Program został uruchomiony w Ośrodku Obliczeniowym Instytutu Technologii i Konstrukcji Budowlanych Politechniki Poznańskiej. Od pewnego czasu jest on szeroko eksploatowany przez pracowników i studentów naszego Instytutu.

DOM HANDLOWY NAUKI Sp. z o.o. POLSKIEJ AKADEMII NAUK - "PROGRAMEX"
Warszawa, ul. Hoża 74, tel. 28-67-76, 29-56-97

Uniwersalny program systemowy ZZU

Oferujemy rewelacyjny program systemowy obsługujący informacje zapisane w pamięciach dyskowych systemu MEKA-400. Program działa na dowolnej wersji systemu operacyjnego SOM-3 (w tym wersjach opracowanych w IŁUW oraz w C.B.W. "Mercomp").

ZZU w porównaniu ze standardowo dostarczanymi programami (LIB, LBU, CAI, CAU, DHP) wyróżnia się bardzo dużą szybkością działania, a wiele bogatszym zestawem możliwości i dyrektyw oraz zwięzła i prosta ich postać.

znajduje zastosowanie we wszelkich operacjach dotyczących pamięci dyskowych. W szczególności może być stosowany do kopiowania dowolnych fragmentów dysków, "podśledzania" informacji na dysku, kontekstowo wyszukiwania wskazanej informacji, modyfikacji wskazanych obszarów dyskowych, wyszukiwania i kopiowania poprawnych modułów binarnych.

Posiada pełną obsługę bibliotek słownikowych (CAI) oraz sekwencyjnych (LIB) w tym obsłudze modułów typu "obraz pamięci".

ZZU oprócz wydobywania i bezpieczeństwa pracy zawiera dodatkowo wiele mechanizmów poszerzających możliwe zastosowanie, jak np. kontrola poprawności transmisji z możliwością ignoracji błędów zapisu/odczytu; kopiowania kaset z dowolnym kluczem adresacji; kalkulator liczb całkowitych podwójnej precyzji; kreowanie sekcji tymczasowych; pętla DO... umożliwiająca wielokrotne wykonanie ciągu dyrektyw ZZU; zmiennie pomocnicze i inne. Program jest odporny na błędy użytkownika oraz ma wbudowaną diagnostykę błędów.

Dla ilustracji podajemy przykładowe parametry ZZU:

- 1) czas kompresji 40 cylindrowej biblioteki słownikowej wraz z kontrolą poprawności programów - 10-14 sek. (nie używana jest tutaj żadna sekcja pomocnicza).
- 2) Pełny listing 40 cylindrowej biblioteki słownikowej wraz z kontrolą poprawności programów - 13 sek.
- 3) Czas kopiowania całej kasety dyskowej (200 cylindry) ok. 30 sek.
- 4) Odszukanie na (końcu) 30 cylindrowej biblioteki sekwencyjnej programu binarnego - 3-4 sek.

Oto przykładowe sekwencje dyrektyw ZZU wraz z ich krótkim opisem:

- a) Tworzymy na sekcji AL1 bibliotekę słownikową. Katalogujemy na pierwszej 30 poprawnych programów binarnych znajdujących się na sekcji AL2 zaczynając od programu FORD:

U AL1; KLESE; 1 AL2; CA*/FORD//; L!

Powyższa sekwencja zakataloguje każdy program pod nazwą równą trzem pierwszym znakom nazwy programu, a następnie wyświetlowana zostanie zawartość biblioteki wraz z informacją o stopniu jej wypełnienia. Czas działania - ok. 35 sek. Analogiczna akcja dla procesora systemowego CAI (jak osobnie wiadomo) trwa "nieco dłużej".

- b) Wypełnienie znakami EDF całej sekcji AL1:

E 40*24 AL1

- c) Łączenie sektorów na sekcji AL2, tzn. ostatnie słowo każdego sektora ma zawierać numer następnego sektora; pozostałe słowa w sektorze wyzerowane. Oto sekwencja:

ZER 40*24 AL2; K=0; DO 40*24 U AL2/K; M 255 K+1; K=K+1

Program ZZU oferowany jest w trzech wersjach: zubożonej, podstawowej, rozszerzonej.

Zamówienia przyjmuje DHN (adres jak wyżej); szczegółowe informacje:



amepol

PRZEDSIĘBIORSTWO

ZAGRANICZNE

George W. Jarnoc
Aleksander Nasielski

USA
Australia

05-311 Dębe Wielkie
Aleksandrówka 20 a
Tlx: 812539 apol pl

Oferta na dostawę sprzętu rozszerzającego możliwości systemu MERA 400

Mamy zaszczyt poinformować Sz. Państwa, że produkujemy moduły i urządzenia umożliwiające rozbudowę systemu MERA-400 i dajemy tym samym szansę dalszej eksploatacji i zwiększenia obszaru zastosowań tego systemu.

Wszystkie nasze wyroby wykonywane są z zachowaniem standardów przyjętych w systemie MERA-400 oraz funkcjonalnie i konstrukcyjnie są przystosowane do przewidywanej przez naszą firmę koncepcji rozbudowy tego systemu w latach następnych.

Produkujemy pamięć operacyjną zrealizowaną na nowoczesnych elementach scalonych 64 K x 1 bit lub 256 K x 1 bit /DRAM/ produkcji zachodniej.

Pamięci oferujemy o pojemności od 64 K słów do 1 MEGA słów z tym, że od połowy 1985 r. oferujemy kolejną wersję pamięci o nazwie MEGA. Jako opcję oferujemy podtrzymanie zasilania pamięci /Back-up/ dla każdej z oferowanych pojemności.

Jednocześnie informujemy, że od drugiego półrocza 1985 roku będziemy produkować moduły procesorów peryferyjnych MULTIX i PLIX zwiększających istotnie możliwości obliczeniowe i zastosowania systemu MERA-400.

Moduł MULTIX jest inteligentnym procesorem peryferyjnym przeznaczonym do dołączenia do 32 wolnych urządzeń peryferyjnych i realizującym protokoły komunikacyjne.

Moduł PLIX jest inteligentnym procesorem peryferyjnym przeznaczonym do dołączenia pamięci masowych. Moduł ten proponujemy z jednostkami sterującymi do następujących dysków:

AMERA 9425

AMERA 9450 - 100 TPI

AMERA 9450 - 200 TPI

BC 5061 /30 MB/

BASF 6185 /Winchester 27,5 MB/

oraz z jednostką pamięci taśmowej TP - 305.

W celu ułatwienia czynności operatorskich i podniesienia niezawodności systemu MERA-400 od drugiej połowy roku 1985 proponujemy moduł inicjatora /INICJATOR/. Moduł ten eliminuje stosowanie czytnika i perforatora przy instalowaniu systemów operacyjnych i testowaniu systemu. W/wym. moduł zrealizowany. Jest na elementach typu EPROM w których zapisane są ściągaczki systemów operacyjnych, programy inicjacji kaset oraz testy pamięci operacyjnej, procesora, przystawki zmiennego przecinka, wybrane programy systemu testów lub programy uzgodnione z użytkownikiem. Uruchomienie odpowiednich programów wymaga dokonania prostych czynności na pulpicie minikomputera MERA-400.

W tym samym czasie oferować będziemy również system operacyjny CROOK-4 dla konfiguracji systemu MERA-400 rozbudowanych o moduły MULTIX, PLIX i MEGA.

W przypadku zainteresowania zamówienia prosimy kierować na adres:

Przedsiębiorstwo Zagraniczne "AMEPOL"
Zakład Elektroniki i Aparatury Medycznej
Plac Żelaznej Bramy 1, 00-136 Warszawa
Informacja tel. 20-34-75, 20-45-05
telex 812539 apol pl

MEGA

Półprzewodnikowa pamięć operacyjna do minikomputera MERA-400

Moduł półprzewodnikowej pamięci operacyjnej MEGA o pojemności do 1 M słów jest przystosowany do minikomputera MERA-400.

Pamięć MEGA może być dostarczona w wersjach:

- 128 K słów / 256 K słów na elementach o pojemności 64 K bitów.

- 512 K słów / 1 M słów na elementach o pojemności 256 K bitów.

Pamięć MEGA może współistnieć z dowolnymi innymi modułami pamięci

zainstalowanymi w systemie minikomputera MERA-400 np. z pamięciami

ferytowych, drutowymi lub innymi pamięciami półprzewodnikowymi.

W module MEGA znajduje się pamięć EPROM o pojemności 4 K słów

z zaprogramowanym inicjalizatorem systemu operacyjnego oraz wstępnym

oprogramowaniem testująco/diagnostycznym systemu minikomputerowego.

Opcjonalnie pamięć MEGA może być wyposażona w układy podtrzymania zawartości pamięci w przypadku krótkotrwałych zaników sieci.

M-400-PP0-1

Półprzewodnikowa pamięć operacyjna do minikomputera MERA 400

Pamięć instalowana jest w szufladzie jednostki centralnej minikomputera obok istniejącej pamięci operacyjnej lub na życzenie klienta w miejsce tejże pamięci lub jako moduł wolostojący. Pamięć półprzewodnikowa może współpracować z pamięcią ferrytową lub może ją zastępować. Ze względu na ulotność informacji w półprzewodnikowych układach pamięciowych, zastosowano bateryjne podtrzymanie zasilania zapewniające ochronę zapisanej informacji przez czas ok. 1 godziny. Pamięć składa się z nast. pakietów : interfejs, sterowanie, pakiety nośnika informacji, pakiety alokacji.

Dane techniczne

- pojemność pamięci : od 64 k do 512 k słów
- długość słowa : 16 bitów + 1 bit parzystości
- czas cyklu zapisu i odczytu : 900 ns
- czas cyklu regeneracji : 450 ns co 16 μ s
- napięcie zasilania : + 5V \pm 10%
- pobór prądu dla pojemności 128 k słów :
6,8 A w czasie normalnej pracy
0,83 A w czasie podtrzymania baterijnego
- czas baterijnego podtrzymania zasilania : ok. 1 godziny

Dane konstrukcyjne

- mikroukłady pamięciowe DRAM o pojemności 64 k x 1 bit produkcji zachodniej
- konstrukcja na półpakietach typu MERA 400 o wymiarach 140 x 300 mm
- łączówki ELTRA 83108401310001
- typy pakietów :
pakiet interfejsu ME JF-400-1
pakiet sterowania ME ST-400-2
pakiet nośnika danych ME DM-400-1
pakiet alokacji ME AL-400-2
- moduł baterijnego podtrzymania zasilania składa się z pojemnika z akumulatorami i pakietu ME BU-400-1 lub zasilacza AMZE 01.

MULTIX

Procesor peryferyjny do wolnych urządzeń zewnętrznych do minikomputera MERA 400

MULTIX jest inteligentnym procesorem peryferyjnym przeznaczonym dla dołączania wolnych urządzeń zewnętrznych do minikomputera MERA-400.

Przesyłanie danych do /z/ urządzenia może odbywać się w sposób równoległy lub szeregowy.

Transmisja szeregowo prowadzona jest w trybie synchronicznym lub asynchronicznym. Znak może składać się z 5, 6, 7, 8 bitów i ewentualnie bitu parzystości / nieparzystości /.

Szybkość transmisji wynosi 150, 300, 600, 1200, 2400, 4800, 14/ 9600 bodów. Linia transmisji szeregowej jest wyposażona w układy nadajników / odbiorników współpracujące z jednym z interfejsów :

- interfejs napięciowy V 24 / EIA-RS 232-C, S 2 /. Maksymalna długość połączenia 30 m

- interfejs prądowy symetryczny / pęta prądowa / o wydajności źródła 12 mA. Maksymalna długość połączenia przy użyciu linii symetrycznej 1000 m
- interfejs prądowy niesymetryczny /- 12 mA / z optoizolacją na wejściu odbiornika. Maksymalna długość połączenia 1000 m.

Parametry transmisji / tryb, format znaku, szybkość / oraz typ interfejsu są ustalane niezależnie dla każdej linii. Procesor MULTIX może obsługiwać maksymalnie 32 linie szeregową transmisji danych.

Za pośrednictwem łącz równoległych przewidziane jest dołączenie następujących urządzeń zewnętrznych :

- czytnik taśmy papierowej CT 2000,
- dziurkarka taśmy papierowej DT 105,
- drukarka mozaikowa DZM 180
- pamięć na dyskach elastycznych SP 45 DE
- pamięć kasetowa PK 1

W procesorze MULTIX znajduje się wspólna dla urządzeń znakowych jednostka sterująca, do której można dołączyć : jeden czytnik CT 2000, jedną dziurkarkę DT 105 oraz dwie drukarki DZM 180. Pozostałe urządzenia / SP 45 DE, PK 1 / posiadają odrębne jednostki sterujące dla każdego z urządzeń. Zwiększenie liczby urządzeń danego typu dołączonych do MERY-400 odbywa się przez zainstalowanie w MULTIXIE dodatkowych jednostek sterujących żądanego typu.

Współpraca procesora MULTIX z minikomputerem MERA 400 polega na pobieraniu i realizowaniu przez procesor rozkazów i poleceń sterujących wysyłanych przez MERE-400 za pośrednictwem rozkazów IN i OUT. Zakończenie realizacji polecenia sygnalizowane jest zgłoszeniem przerwania.

Zakres poleceń sterujących obejmuje logiczne dołączanie i odłączanie urządzeń znakowych i linii transmisji szeregową do / od MERY-400 oraz transmisję pojedynczych znaków lub bloku znaków między pamięcią operacyjną MERY a urządzeniem zewnętrznym. Logiczne dołączanie linii transmisji szeregowej ma na celu ustalenie parametrów transmisji oraz rodzaju protokołu liniowego / o ile taki jest stosowany na danej linii /. Podczas logicznego dołączania urządzenia przesyłającego dane równoległe określa się sposób współpracy z urządzeniem / czy transmisja odbywa się „znak po znaku” czy w sposób blokowy /. Procesor MULTIX może prowadzić wstępną obróbkę danych przesyłanych do / z urządzenia. Obróbka danych może mieć na celu : wyszukiwanie, usuwanie, dostawianie określonych znaków sterujących, konwersję kodu, realizację funkcji związanych z zastosowanym protokołem liniowym.

Systemy operacyjne obsługujące w/w wyrób w opracowaniu.

PLIX **Procesor peryferyjny do pamięci masowych do minikomputera MERA 400**

PLIX jest inteligentnym procesorem peryferyjnym przeznaczonym dla dołączania szybkich pamięci masowych do minikomputera MERA-400.

Przewiduje się współpracę procesora Plix z następującymi typami pamięci masowych :

- pamięć dyskowa MERA 9425, 9450 o pojemności 5 Mbajtów. Zachowana jest wymiennność kaset dyskowych użytkowanych za pośrednictwem procesora Plix z kasetami użytkowanymi poprzez standardowy moduł pamięci dyskowej MPD 9425-400, 9450-400.
- pamięć dyskowa EC 5061 / IBM 2314 / o pojemności 30 M bajtów. Przewiduje się późniejsze dwukrotne zwiększenie pojemności pamięci EC 5061 /do 60 MB/ przez zastosowanie metody zapisu MFM zamiast obecnie stosowanej FM.
- pamięć taśmowa PT 305 z możliwością metod zapisu NRZI oraz PE. Pamięć PT 305 jest dołączana do procesora Plix za pośrednictwem formatera FRPT-305.

Dla każdego typu pamięci w procesorze Plix znajduje się wydzielona jednostka sterująca. Do poszczególnych jednostek sterujących mogą być dołączone : 1 do 4 pamięci MERA 9425 /9450/ połączonych szeregowo.

1 do 4 pamięci EC 5061 połączonych szeregowo, 1 do 4 pamięci PT 305 połączonych szeregowo przez formater FRPT-- 305. W celu zwiększenia liczby pamięci dołączonych do MERY-400 / ponad 4 urządzenia danego typu / należy zainstalować w procesorze Plix dodatkowe jednostki sterujące żądanego typu.

Przewiduje się możliwość dołączania innych, nie wymienionych wyżej pamięci masowych, np. dysków typu Winchester lub bardzo szybkich linii transmisji szeregowej. Maksymalna szybkość transmisji danych może wynosić 12 M bitów/sek.

Współpraca procesora PLIX z minikomputerem MERA-400 polega na pobieraniu i realizowaniu przez procesor PLIX rozkazów i poleceń sterujących wysyłanych za pośrednictwem rozkazów IN i OUT przez MERE-400. Zakończenie realizacji polecenia sygnalizowane jest zgłoszeniem przerwania. Zakres poleceń sterujących obejmuje wykonywanie czynności związanych z pozycjonowaniem głowic dysku, szukaniem bloku o zadany numerze, transmisją bloku danych między pamięcią masową a pamięcią operacyjną MERY-400.

Przewiduje się zmniejszenie średniego czasu dostępu do pamięci dyskowej przez optymalizację ruchu zespołu głowic. Transmisja bloku danych pomiędzy pamięcią masową a operacyjną odbywa się za pośrednictwem buforów w procesorze PLIX. Zapobiega to nienadżądaniu transmisji oraz przesyłaniu błędnie odczytanych bloków do pamięci operacyjnej. W przypadku wystąpienia błędu odczytu lub pozycjonowania automatycznie następuje próba ponownego przeprowadzenia operacji w sposób poprawny.

Systemy operacyjne obsługujące w/w wyrób w opracowaniu.

Ponadto AMEPOL oferuje :

AMPAC-11 Półprzewodnikowa pamięć operacyjna
do procesorów PDP 11/SK.

MEDH.915.1 Półprzewodnikowa pamięć operacyjna
do procesorów NOVA 1200, ES-P3; DCC-116.

ODDM 05/2M i ODDM-256 K Półprzewodnikowe pamięci
operacyjne do komputera ODRA 1305.

R 32.2MB Półprzewodnikowa pamięć operacyjna do
komputera RIAD 32.

mercomp

Sp.z.o.o.

CENTRUM BADAWCZO - WDROŻENIOWE

04-094 Warszawa ul. Pozejł 19 Tel. 12 91 30 Tlx 81 58 57 zptmp pl

Bank PKO S.A. nr. rachunku 801031-131-1113-9000070 | IDENT. GUS

podrzednictwem III OM MBP W-wa nr. rachunku 1036-7878-138 | 1094819

- AUTOMATYKA I SYSTEMY
- KOMPUTEROWE
- DORADZTWO
- PROJEKTOWANIE
- POMIARY I AUTOMATYKA
- SPRZET KOMPUTEROWY
- I OPROGRAMOWANIE
- KOMPLETACJA
- MONTAŻ
- ROZRUCH
- OBSŁUGA
- SERWIS

Oddział Warszawski - Informatyka

Warszawa, ul. Rozbrat 32 m 6

mgr inż. Wojciech Szanser tel. 29-93-49

mgr inż. Stanisław Gałązka tel. 25-56-40

Kompetentnych i pełnych informacji o oprogramowaniu udzielają pracownicy Zespołu Oprogramowania Podstawowego MERA-400.

WARSZAWA, ul. Armii Ludowej 16, pok. 232
tel. 25-92-80

mgr inż. Krzysztof Bryczkowski
dr Krzysztof Miziołek
mgr Zbigniew Swirski

mgr Jerzy Majewski tel. 23-76-01 w. 297
mgr Krzysztof Wagner

```

#####      ##      ##      ##      ##      ##      ##
## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ## ##
##
#####

```

Centrum Badauczo-Wdrozeniowe "MERCOP" Sp. z o.o.
 04-994 Warszawa, Foezji 19, tel. 12-91-30

```

##### -      #####      #      #####
#####      #####      #####      ##      #####      ##
#####      #####      #####      #####
#####      #####      #####      #####
#####      #####      #####      #####
#####      #####      #####      #####
#####      #####      #####      #####
#####      #####      #####      #####
#####      #      #####      ##      #####      TM

```

----- MULTI-ACCESS-TASK MONITOR

Monitor wielodostepu MAT (Multi Access Task) sluzyc do budowania systemow konwersacyjnych pracujacych wspolbieznie z wieloma koncowkami /terminalami/. Z kazda koncowka zwiazany jest jeden proces wykonywania programu. MAT rozdzielajac czas pracy maszyny przydzielony zadaniu rownomiernie pomiedzy procesy, stwarza efekt rownoczesnego ich dzialania.

MAT zawiera rowniez mechanizm synchronizacji procesow. Dzieki tym wlasciwosciom jest mozliwe w oparciu o wspolne dane wykonywanie z roznych terminali tych samych lub roznych czynnosci jednoczesnie np.: wprowadzanie danych, wyszukiwanie informacji, sporzadzanie zestawien, itp.

Program uzytkowy moze byc napisany w jezyku SIMBOL, FORTRAN lub macroassembler.

Monitor MAT jest generowany w zaleznosci od konfiguracji sprzetowej i programowej.

Monitor wielodostepu MAT funkcjonuje jedynie pod systemem operacyjnym SOM-3.MAT.

ZASADY DZIAŁANIA MATA

Monitor MAT wykorzystuje czas, w którym proces oczekuje na zakończenie zainicjowanej przez siebie instrukcji wejścia-wyjścia, na działanie innych procesów. Z tego powodu wszystkie transmisje znakowe powinny być wykonywane za pośrednictwem odpowiednich procedur wejścia-wyjścia MATA. Każdy współbieżny proces związany jest z jednym terminalem. Przełączenie tzn. przekazanie czasu maszyny między procesami odbywa się w trakcie transmisji znakowych i znacznej części innych procedur MATA. Czas maszyny przekazywany jest wtedy pierwszemu w kolejce procesowi, który nie oczekuje na zakończenie transmisji lub nie jest celowo zawieszony. Ochrona danych i programów przed zakłóceniami powodowanymi współbieżnie wykonywanymi procesami jest realizowana automatycznie i polega na składowaniu ciał procesów (zawartości pamięci programu i danych) w pamięci operacyjnej lub dyskowej na czas działania pozostałych procesów. Wymiana ciał procesów dokonywana jest w trakcie przełączania procesów.

Rozpoczęcie pracy współbieżnej w programach nadzorowanych przez monitor MAT możliwe jest dopiero po wykonaniu procedury MULTISTART. Do momentu wywołania tej procedury w programie działa tylko jeden proces, a odwołanie do innych procedur MATA (z wyjątkiem procedury TYPHON) spowoduje odrzucenie zadania. Procedura MULTISTART zapisuje ciało pierwszego procesu na sekcji dyskowej za pośrednictwem strumienia WRK. Zawieszony w ten sposób stw. programu jest odtwarzany w momencie dołączenia każdej kolejnej koncówki a proces z nią związany rozpoczyna działanie od pierwszej instrukcji za wywołaniem procedury MULTISTART.

Gdy składowanie ciał procesów dokonywane jest w PAO to dołączenie nowego procesu wiąże się z przydzieleniem mu odpowiedniej ilości pamięci przez pobranie tej z puli wolnych ramek systemu do puli zadania. Przełączenie polega na wylczeniu z przestrzeni adresowej ramek podanego procesu i włączenie ramek następnego. Odłączenie procesu (procedura ENDSESSION) powoduje zwrócenie zajmowanej przez proces pamięci do puli wolnych ramek systemu.

Gdy składowanie ciał procesów dokonywane jest w pamięci dyskowej to przełączenie polega na zainicjowaniu na dysku ciała jednego procesu i wczytanie ciała następnego procesu. W tym przypadku sekcja dyskowa związana ze strumieniem WRK musi być odpowiednio duża by pomieścić ciała wszystkich procesów oraz początkowy stw. programu.

Ze względu na odseparowanie procesów i związaną z tym niedostępność do danych innego procesu, MAT wyposażony jest w aparat umożliwiający przekazywanie informacji pomiędzy procesami oraz synchronizację procesów. Do tych celów służą dostępne dla wszystkich procesów zmienne MATA zwane semaforami oraz odpowiedni zestaw procedur.

Przetwarzanie programów użytkowych współpracujących z MATEM (kompilacja, "asemblacja", łączenie) realizowane jest niezależnie od MATA. Wskazane jedynie na strumieniu bibliotecznym LB lub USL "warstwy pośredniczącej" zapewniającej komunikację programu użytkowego z

monitorem MAT. Wywoływanie procedur MATa realizowane jest przy pomocy instrukcji CALL (w SIMBLU i FORTRANie) lub instrukcji RJ0,4, (nazwa) z łącznikiem zawierającym adresy parametrów (w assemblerze).

Monitor MAT jest słownym overlayem programu użytkowego. Program taki musi być wywoływany za pośrednictwem zadania systemowego SUPERVISOR dyrektywa:

PGM (nazwa-MATa) [(nazwa-zadania)]

W przypadku gdy dany program nie jest aktualnie wykonywany, zostanie załadowany Job-Control ZEK, by umożliwić dokonanie pewnych czynności wstępnych (np. przydział strumieni). MAT zostanie załadowany po podaniu dyrektywy:

RUN

MAT bezpośrednio po starcie ładuje program użytkowy i przekazuje mu sterowanie. Od tego momentu program jest wykonywany przez jeden proces związany z końcówką inicjująca wykonywanie programu.

W przypadku gdy dany program jest aktualnie wykonywany, dyrektywa:

PGM (nazwa-MATa) [(nazwa-zadania)]

nie spowoduje zainicjowania nowego zadania w systemie, lecz spowoduje uruchomienie w działającym programie nowego procesu związanego z końcówką, z której podano dyrektywę PGM. Parametr (nazwa-zadania) jest w tym przypadku ignorowany.

Zakończenie działania procesu (procedura ENDSESSION) spowoduje odłączenie końcówki od zadania i przekazanie jej pod kontrolę SUPERVISORA. Zakończenie działania ostatniego procesu wiąże się z zakończeniem pracy zadania (jak w dyrektywie BYE Job-Controla).

WERSJE MONITORA MAT

Monitor MAT jest generowany w zależności od określonych potrzeb użytkownika. Pozwala to na utworzenie takiej wersji monitora, która najlepiej odpowiada potrzebom systemu użytkowego, z którym współpracuje.

W czasie generacji definiowana jest konfiguracja sprzętowa, podawany jest zestaw wykorzystywanych funkcji monitora, określany jest język programowania, w którym jest napisany system użytkowy oraz inne cechy charakteryzujące wersje monitora.

W ramach definiowania konfiguracji sprzętowej określa się:

- ilość procesów współbieżnych;
- strumienie wejścia-wyjścia przyporządkowane poszczególnym procesom;
- ewentualne przywiązanie procesów do konkretnych końcówek;
- ewentualne wskazanie końcówki startującej program;
- strumienie nazwy urządzeń i opcje tablicy UFT dla urządzeń dodatkowych.

Okreslenie języka programowania systemu użytkowego (SIMBOL, FORTRAN) jest niezbędne z uwagi na różnice w postaci przekazywanych parametrów procedur w tych językach. Programy użytkowe pisane w macroassemblerze powinny współpracować z fortranowską wersją MATA.

Podczas generowania monitora określa się ilość semaforów.

W definicji monitora określa się nazwę programu użytkowego, ilość ramek pamięci, konieczna dla jego prawidłowego działania i określa się skąd ma być ładowany (strumień, ew. sekcja, pakiet dyskowy).

Definicja MATA zawiera również specyfikacje wszystkich procedur, które mają być do niego dołączone. Jedynie procedury MULTISTART i ENDSERION dołączone są zawsze i nie wymagają specyfikacji.

LISTA PROCEDUR

Niżej przedstawiamy listę procedur wraz z parametrami. Procedury dokonujące przełączenia oznaczono znakiem @.

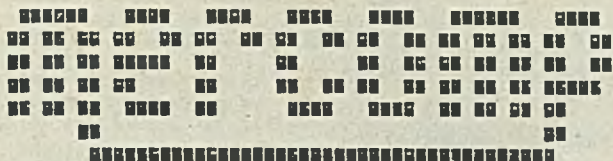
Znaczenie parametrów:

- p** - numer procesu (zmienna lub stała całkowita);
- v** - zmienna lub stała całkowita;
- text** - w SIMBOLU - zmienna lub stała tekstowa, w FORTRANIE - dwa parametry: nazwa tablicy lub stała tekstowa oraz ilość znaków;
- can** - w SIMBOLU - zmienna lub stała typu can; w FORTRANIE - dwa parametry: nazwa tablicy lub stała can oraz ilość znaków;
- u** - nazwa strumienia; w SIMBOLU - zmienna lub stała typu can; w FORTRANIE dowolna dana zawierająca tekst can;
- r** - zmienna logiczna;
- n** - numer semafora (zmienna lub stała całkowita);
- typ** - typ monitora (zmienna lub stała całkowita = 0, 1 lub 2);
- wzt** - wskaźnik zakończenia transmisji (zmienna całkowita);
- q** - ilość ostatnio przetransmitowanych znaków (zmienna całkowita).

Procedury wejścia-wyjścia z przedrostkiem D dokonują transmisji z koncówką lub urządzeniem opcjonalnym w zależności od przedziału dokonane przez procedury INDEVICE i OUDEVICE. Procedury wejścia-wyjścia bez przedrostka D komunikują się zawsze z koncówką.

Procedura	Znaczenie
@ MULTISTART (wzt, q)	Rozpoczęcie wykonywania procesów współbieżnych.
@ ENDSERION	Koniec pracy procesur, odłączenie koncówki.
@ HOLD (o)	Zawieszenie procesu.

⊙ RESTART (p)	Wznowienie procesu.
STOP	Koniec pracy zadania, zwolnienie wszystkich koncowek.
⊙ EDJIINT (v, t)	Czytaj liczbe całkowita.
⊙ EDJOINT (v, t)	Pisz liczbe całkowita.
⊙ EDJITEX (text, t)	Czytaj tekst w kodzie ISO.
⊙ EDJOTEX (text, t)	Pisz tekst w kodzie ISO.
⊙ EDJICAN (can, t)	Czytaj tekst w kodzie CAN.
⊙ EDJOCAN (can, t)	Pisz tekst w kodzie CAN.
INDEVICE (u)	Przydziel urządzenie opcjonalne wejsciowe.
OUDEVICE (u)	Przydziel urządzenie opcjonalne wyjsciowe.
ONFULL	Włącz sygnalizacje wypełnienia ekranu.
OFFULL	Wyłącz sygnalizacje wypełnienia ekranu.
⊙ TAKE (u, r)	Zarezerwuj urządzenie.
⊙ TAKEMAIT (u)	Czekaj na zwolnienie i zarezerwuj urządzenie.
⊙ GIVE (u)	Zwolnij urządzenie.
⊙ GIVEALL	Zwolnij wszystkie urządzeniu.
⊙ SEMPUT (v, n)	Przypisz n-temu semaforowi wartosc v.
⊙ SEMGET (v, n)	Przypisz zmiennej v wartosc n-tego semafora.
⊙ SEMWAIT (v, n)	Czekaj do momentu gdy wartosc n-tego semafora bedzie rowna v.
PROCESS (p)	Przypisz zmiennej p numer aktywnesnu procesu.
⊙ EMPTY	Wykonaj przelaczenie.
EXCLUSIVE	Zawies wszystkie pozostale nie zawieszzone procesy.
EQUAL	Wznow wszystkie procesy zawieszzone przez EXCLUSIVE.
TYPMON (p, tno)	Deklaracja typu monitora - terminala zwiazanego z procesem p.
MESSAGE (text, p)	Przeslij komunikat na koncowke procesu p.



Centrum Badawczo-Wdrożeniowe "MERCAMP" Sp. z o.o.
04-974 Warszawa, Poczta 19, tel. 12-91-30

C.B.W. "MERCAMP" (Jednostka gospodarki uspołecznionej) - Zespół Oprogramowania Podstawowego MERY-400 oferuje użytkownikom tego minikomputera atrakcyjne oprogramowanie:

- Ulepszona wersja systemu operacyjnego SOM-3.
- Nowy język sterowania zadaniem (Job-Control).
- Kompilatory języków programowania.
- Uzupełniające oprogramowanie narzędziowe.

Całe przedstawione tutaj oprogramowanie oparte jest na rozszerzonej wersji systemu operacyjnego SOM-3 opracowanej w naszym zespole i nazywanej dalej SOM-3.M.

1) System operacyjny SOM-3.M

System operacyjny SOM-3.M jest rozszerzoną, ulepszoną wersją systemu operacyjnego SOM-3. Starano się w nim pozbyć wszelkich niedosadności i błędów zawartych w systemie dostarczonym przez producenta minikomputerów MERA-400 oraz innych wersjach systemu SOM-3. Szczególny nacisk położono w nim na pracę wielozadaniową, możliwość wykorzystania dużej puli pamięci operacyjnej oraz na pracę z procesorami periferyjnymi.

Do podstawowych cech omawianego systemu zaliczyć należy:

- Wielozadaniowość

System posiada pełne możliwości pracy wielozadaniowej (jednoczesna, interakcyjna praca wielu użytkowników komunikujących się z minikomputerem przy pomocy terminali) pozwalającej w sposób optymalny wykorzystywać moc obliczeniową procesora.

Zadania mają zapewniony - w odróżnieniu od innych, znanych nam, wersji systemu SOM-3 - sprawiedliwy podział czasu dostępu do procesora, nie faworyzujący nigdy żadnego z nich. Istnieje możliwość uprzywilejowania (w sensie czasu dostępu do procesora) określonego

zadania użytkowego, bądź stworzenia zadania mniej uprzywilejowanego (tzw. praca w tle). Praca wielozadaniowa możliwa jest na wszystkich końcówkach (terminalach) dostosowanych do systemu HERA-400, w tym m.in. na monitorach 7932, 7933, 7950, KSR oraz monitorach zleżnych 7910 (przez jednostkę grupową 7905).

- Zarządzanie końcówkami.

Specjalne komunikacyjne zadanie systemowe SUPERVISOR pozwala na niezależnienie zadań od końcówek w ten sposób, że użytkownik na początku pracy wskazuje jakie zadanie ma pracować na danym terminalu.

SUPERVISOR jako zadanie nadrzędne steruje również pracą zadań wielodostępnych rezerwując dla nich odpowiednie terminale systemu.

Wersja systemu umożliwiająca te funkcje nazywa się **SOH-3.MAT.**

- Zarządzanie pamięcią operacyjną

W systemie istnieją łatwe mechanizmy dostępu i optymalnego zarządzania całą (niezależnie od jej wielkości) pamięcią operacyjną minikomputera. Każde z zadań posiada własny obszar pamięci operacyjnej z możliwością dynamicznej zmiany jej wielkości w zależności od aktualnych potrzeb użytkowników. Maksymalna wielkość pamięci zadania wynosi 64 K słów, minimalny 4 K słów. Istnieje możliwość wirtualnego dostępu (także w Fortranie) do pamięci operacyjnej przekraczającej rozmiary danej zadania do 0.5 M słów. Jest to szczególnie przydatne dla konfiguracji o dużym zasobie pamięci. Systemowe biblioteki zawierają m.in. procedury pozwalające w wydajny sposób korzystać z całej, dostępnej w systemie pamięci operacyjnej.

- Kompatybilność

System SOH-3.M jest rozszerzeniem standardowej wersji systemu SOH-3 w wersjach BASIC i FNC. Programowanie utworzone pod tymi systemami jest w naturalny sposób akceptowane przez SOH-3.H. W szczególności możliwy jest łatwy dostęp do kaset dyskowych FNC oraz kaset ze standardowym podziałem na sekcje. Oferowany system zapewnia także kompatybilność z wersją systemu SOH-3 opracowaną w IIUW.

- Handlers

System posiada możliwość współpracy ze wszystkimi urządzeniami peryferyjnymi minikomputera HERA-400. M.in. zawiera mechanizmy obsługi pamięci dyskowej EC 5061 (25 MB), monitorów 7950, drukarek wierszowych DW-3, taśm magnetycznych, dysków elastycznych, jednostki grupowej 7905 itd. W odróżnieniu od standardowej wersji SOH-3 ulepszone zostały programy obsługi (handlers), wielu urządzeń peryferyjnych. Zmiany te wyraźnie poprawiają komfort pracy pod tym systemem.

Oferujemy również handler inteligentnego procesora peryferyjnego przeznaczony do łączenia szybkich pamięci zewnętrznych (dyski i taśmy) - **PLIX** (w opracowaniu) oraz peryferyjnego procesora komunikacyjnego do łączenia urządzeń znakowych - **MULTIX**.

- Ekstrakody

Do systemu dołączono szereg nowych ekstrakodów rozszerzających możliwości programów użytkowych. W szczególności istnieje możliwość modyfikowania niektórych elementów systemu ustalanych dotychczas w SOM-3 na etapie generacji, jak np.: aktualnej pamięci zadania, położenia sekcji dyskowych (w tym roboczych), czasu zesora, zwiększenia/zmniejszenia ilości niektórych stałych zasobów zadania, itp.

Poprawiono i rozszerzono wiele standardowych ekstrakodów usuwając w ten sposób dokuczliwe ich błędy lub braki.

- Zadanie komunikacji

Modyfikacje handlerów urządzeń spowodowały istotne zmniejszenie częstotliwości występowania "OFF-LINE". Jeżeli jednak przypadek taki występuje (np. wyłączone urządzenie) to zmodyfikowane zadanie komunikacji (lokalne) zaszyta się automatycznie na odpowiedni terminal.

Ponadto dołączono wiele nowych dyrektyw (np. GO, SOM, PAS, DDR) oraz powiększono ilość informacji przekazywanych przez dotychczasowe dyrektywy raportujące (np. TYF, REF).

2) Job-Control ZZK

Job-Control ZZK jest całkowicie nowym językiem sterowania zadaniem stanowiącym istotne rozszerzenie w stosunku do innych analogicznych, znanych nam języków minikomputera HERA-400. Pracuje na bazie systemu operacyjnego SOM-3.M. Jego podstawowa cecha jest zredukowanie do niezbędnego minimum potrzeby znajomości cech systemu operacyjnego. Bogaty repertuar dyrektyw (zadawalający również prostą obsługę systemową) zapewnia wygodną obsługę kompilatorów (Mac (Mca), Gass, Fortran, Pascal, Simbul, Loglan), programów użytkowych oraz pełnej informacji o zasobach danego zadania.

ZZK przystosowany jest do pracy wielozadaniowej z uwzględnieniem innych cech systemu operacyjnego jak np. dynamiczna obsługa pamięci operacyjnej zadania, pełny dostęp do sekcji dyskowych oraz pakietów FMC itd.

ZZK posiada wbudowany podsystem zarządzania zawartością dyskowych kaset wymiennych. Logiczna struktura pakietu zawiera podział na biblioteki, które mogą być utożsamiane z użytkownikami danego pakietu. Z biblioteka związana jest pula jej zbiorów (plików) identyfikowanych, tak jak biblioteki, przez nazwę. Biblioteki pakietu zawierają zbiory różnych typów. Istnieją cztery standardowe typy zbiorów:

- S - source (programy źródłowe, teksty)
- B - binary (programy gotowe do wykonania)
- L - library (procedury, biblioteki procedur do linkowania)
- M - macro (makrodyrektywy)

Użytkownik może również definiować własne typy zbiorów.

Organizacja kasety wymiennej jest zatem w ZZK wielopoziomowa

struktura dzwiewiastą: pakiet-biblioteka-tytuł zbioru-zbiór.

ZZK posiada bogaty repertuar dyrektyw umożliwiających operacje na zbiorach. Programy systemowe (np. kompilatory, edytor) uruchamiane przez dyrektywy ZZK posiadają, z punktu widzenia użytkownika, bezpośredni dostęp do zbiorów kasety wymiennej. Istnieją zabezpieczenia na wypadek przypadkowego zamazania części kasety. Zapewniono także możliwość mieszanej organizacji kasety wymiennej w taki sposób by jej część była dostępna w tradycyjny sposób za pomocą sekcji dyskowych. Fizyczna struktura pakietu dyskowego jest również kompatybilna z fizyczną strukturą kasety dostępnej w podsystemie FLS opracowanym w IIUW. Istnieją łatwe mechanizmy przechodzenia z pracy z kasetami FMC na proponowany system zbiorów.

Z ZZK może współpracować podsystem rejestracji czasu pracy użytkowników. Podsystem ten umożliwia dostęp do maszyny tylko uprawnionym użytkownikom.

Do podstawowych informacji rejestrowanych w tym systemie należą: nazwa użytkownika, jego aktualne hasło, godzina i data rozpoczęcia oraz zakończenia sesji (z możliwością zapamiętania 120 ostatnich sesji danego użytkownika), globalny czas pracy użytkowników, itd.

W fazie prób znajduje się nowy, uniwersalny system zbiorów dyskowych dostosowany do wszystkich rodzajów pamięci dyskowych (w tym typu WINCHESTER).

3) Kompilatory

Razem z dostarczonym systemem operacyjnym oferujemy - specjalnie dostosowane przez autorów do naszego oprogramowania - kompilatory języków programowania: Simbol, Mca, Gass, Loylan-62, Pascal.

SIMBOL - jest językiem programowania przeznaczonym głównie dla przetwarzania danych. Charakteryzuje go:

- prosta, strukturalna składnia;
- wygodne operacje na tekstach;
- bogata struktura zbiorów danych (zbiory indeksowane, bezpośredniego dostępu, sekwencyjne, hierarchiczne);
- zależna i niezależna kompilacja procedur.

Jest on znakomitym narzędziem do pisania programów obsługujących różnego typu bazy danych.

W opracowaniu znajduje się SIMBAD - wielodostępny, konwersacyjny program obsługi bazy danych opartej na SIMBOLowej strukturze zbiorów danych.

- Pamięć operacyjna potrzebna dla kompilatora:
- 20k słów
- Pamięć operacyjna obsługiwana przez programy w SIMBOLu:
- 32k słów.

MCA - nowa, ulepszona wersja standardowego makroasemblera MAC. W odróżnieniu od starej wersji charakteryzuje się kilkakrotnie większą szybkością działania oraz rozwinięty, mocniejszy aparat makrodyrektyw.

- Pamięć operacyjna potrzebna dla kompilatora:
20k słów
- Pamięć operacyjna obsługiwana przez programy w MACu:
64k słów (statycznej - w zadaniu) + wolna (wirtualna) pamięć w systemie.

GASS - jest asemblerem zrealizowanym wg. całkowicie nowej koncepcji. Mnemoniczne kody instrukcji zostały wyeliminowane i pozwalają na naturalne i bardzo łatwe zapamiętanie ich nazw z faktycznie wykonywaną operacją. Składnia drugiego argumentu instrukcji została ujednoczona: stowaś pośredniości jest to jednoznacznie określony składnia argumentu.

Jezyk ten w niektórych swoich konstrukcjach przypomina jezyk programowania wyższego rzędu. Jest również wyposażony w makroaparat oraz warunkowa translacje fragmentów kodu. GASS jest kilkakrotnie szybszy od standardowego makroasemblera MAC.

W skład wyposażenia nowego asemblera wchodzi także program FMS służący do formatowania tekstów źródłowych napisanych w GASSie.

- Pamięć operacyjna potrzebna dla kompilatora:
20k słów
- Pamięć operacyjna obsługiwana przez programy w GASSie:
64k słów (statycznej - w zadaniu) + wolna (wirtualna) pamięć operacyjna w systemie.

LOGLAN-82 - jest nowym językiem programowania wysokiego poziomu. Napisany został dla nowej generacji maszyn cyfrowych. Stanowi on, uniwersalne, bardzo silne narzędzie programowania; przodatkę zwłaszcza do łatwego zapisania skomplikowanych algorytmów strukturalnych o charakterze raczej nienumericznym.

LOGLAN-82 posiada bardzo bogaty zestaw narzędzi umożliwiających definiowanie używanych (także hierarchicznych) typów danych oraz ułatwiających programowanie strukturalne. Jezyk ten znakomicie nadaje się do celów symulacyjnych. Posiada mechanizmy dynamicznego organizowania pamięci operacyjnej z programowa kontrola jej wykorzystania. Wszystkie obiekty strukturalne (np. tablice, procedury, funkcje, współprogramy) tworzone są dynamicznie i mogą być kasowane (z odzyskaniem zajmowanego przez nie obszaru pamięci), gdy nie są już potrzebne.

- Pamięć operacyjna potrzebna dla kompilatora:
20k słów
- Pamięć operacyjna obsługiwana przez program w LOGLANie-82:
64k słów

PASCAL - wersja kompilatora implementuje duży zespół elementów tego powszechnie znanego języka programowania, którego omawiać nie trzeba.

- Famiel operacyjna potrzebna dla kompilatora:
40k słów
- Famiel operacyjna obsługiwana przez programy w PASCALu:
64k słów

4) Nowe oprogramowanie systemowe

Oferujemy również nowe wersje tzw. procesorów systemowych: EDI, CAT, LIB o nazwach odpowiednio: EDD, CAD i LBD.

5) Modyfikator systemu

Specjalny program pozwalający wyprodukować nową binarną wersję systemu operacyjnego (w oparciu o podstawową wersję SOH-3.M lub SOH-3.MAT dostarczoną przez C.B.W. "Hercomp") co w wielu wypadkach zastąpi konieczność generacji systemu.

6) Oprogramowanie użytkowe

Niezależnie od zespołu zajmującego się oprogramowaniem systemowym istnieje zespół z dużym doświadczeniem w tworzeniu oprogramowania użytkowego, głównie w zakresie: księgowość, płace, zastosowania medyczne, zarządzanie, itp.

7) Oprogramowanie narzędziowe

Nasza oferta pragniemy zakończyć omówieniem oprogramowania pomocniczego, pozwalającego w wygodny sposób tworzyć oprogramowanie użytkowe, dokumentację, opisy, dane, itd.

EDM - nowoczesny edytor tekstu przewyższający o klasę wszystkie inne edytory minikomputera HERA-400.

Fundamentowe cechy tego programu to bezpośredni i błyskawiczny dostęp do dowolnego fragmentu opracowywanego tekstu oraz silne dyrektywy, pozwalające przy minimalnym nakładzie pracy wykonywać

złożone modyfikacje na całym lub wybranych fragmentach tekstu.

Warto zaznaczyć, że podstawowe mechanizmy zawarte w programie EDM są możliwe do natychmiastowego opanowania przez początkującego użytkownika.

EDS - nowoczesny edytor ekranowy, przystosowany do współpracy z Jednostką grupowa 7905.

FRI - uniwersalny program do listowania zbiorów tekstowych. FRI posiada możliwość wielokolumnowego listowania całego lub fragmentu tekstu z uwzględnieniem numeracji linii, nasłówek oraz ewentualnie małych liter.

LSC - program do szybkiego, wielokolumnowego listowania sYownikowych bibliotek binarnych utworzonych pod standardowym programem CAT lub CAD.

LLB - program do szybkiego, wielokolumnowego listowania sekwencyjnych bibliotek binarnych.

FMR - uniwersalny formater dla programów źródłowych napisanych w języku MCA (MAC) lub FORTRAN.

DOC - program sYuży do łatwego i szybkiego nadawania tekstom postaci wydawniczej. W szczególności ten dokument przystosowany został przy pomocy programu DOC, niech więc będzie on ilustracją możliwości DOCa.

GCP - program do bardzo szybkiego kopiowania sekcji dyskowych.

Programy LOGLAN, PASCAL, GASS, EDM, EDS, FMR, DGC opracowane zostały w IIUW i specjalnie dostosowane przez autorów do naszego systemu.

Kompetentnych i pełnych
informacji o wyżej wymienionym
oprogramowaniu udziela pracownicy
Zespołu Oprogramowania
Podstawowego MERA-400
C.B.W. "Mercomp" Sp. z o.o. :
Warszawa, Armii Ludowej 16 pok. 232
tel. 25 92 60

OFERTA FIRMY "Z E K O M"

Zakład Elektroniki Komputerowej "ZEKOM" jest firmą prywatną specjalizującą się w produkcji monitorów i terminali ekranowych oraz urządzeń i wyposażenia wspomagającego. Na wszystkie produkowane urządzenia udzielamy rocznej gwarancji. Oferujemy również serwis pogwarancyjny w terminie dwutygodniowym.

Obecnie produkujemy następujące urządzenia:

- monitor ekranowy MV 1664,
- terminal nadawczo-odbiorczy MV 1664-KSR,
- terminal odbiorczy MV 1664-RO,
- pakiet sprzęgu SV 24/400,
- sprzęgi SM 11, SM 12, SM 21,
- klawiatury KF 1664
- uchwyt UM 1.

W IV kwartale b.r. rozpoczynamy produkcję terminala nadawczo-odbiorczego MV 2580.

Najważniejsze dane techniczne oraz informacje o zastosowaniu i możliwym wyposażeniu poszczególnych urządzeń można przedstawić następująco:

MV 1564 - monitor ekranowy odbiorczy:

- pojemność ekranu 1024 znaki /w 16 wierszach po 64 znaki/,
- kineskop M31-310GH /12", luminofor P31/,
- obudowa telewizora VELA 203,
- interfejs równoległy o strukturze analogicznej jak w interfejsie drukarki DZM 180, lecz w odwrotnej konwencji sygnałów /logika dodatnia/ obejmujący 7-bitową szynę danych wejściowych + 2 linie sterujące - działający według reguły handshaking'u,
- ilość obsługiwanych znaków: - wyświetlanych 96 /małe i duże litery, symbole arytmetyczne i inne/
kod znaków ASCII - specjalnych 8/NUL, BEL, BS, HT /przesunięcie znacznika o jedną pozycję w prawo bez zapisu/, LF, FF, VT /może pełnić funkcję znaku FF/ CR/
- znacznik nieadresowalny porusza się sekwencyjnie w najniższym wierszu,

- automatyczne przeniesienie zapisu do następnego wiersza po przekroczeniu pojemności danego wiersza,
 - szybkość do 100 000 znaków/s dla znaków wyświetlanych,
 - typowe zastosowanie: oszczędzająca papier współpracująca z drukarką DZI 180; końcówka systemu mikroprocesorowego z układem PIA /np.8255/ na wyjściu.
 - wyposażenie dodatkowe:
 - sprzęgi SM,
 - uchwyt UM-1 mocujący obrotowo monitor do drukarki DZM 180
 - uchwyt UM-21 lub podstawa UP-1 w przypadku użycia monitora jako urządzenia wolnostojącego,
 - przystawka funkcyjna UF-1 do sterowania stronicowaniem oraz spowalnianiem /do szybkości zbliżonej do szybkości drukarki DZM 180 czyli 180 znaków/s,
 - sprzęgi SM umożliwiające podłączenie monitora ekranowego MV 1664 do następujących urządzeń:
 - SM-11 - do drukarek DZM 180, DZM 180/325, monitora technicznego EC 7086 /konsola operatora systemu RIAD-32/,
 - SM-12 - do terminala DZM 180-KSR, monitorów technicznych DZM 180/05 i DZM 180/25 /konsole operatora systemów ODRA 1305 i 1325/-
 - SM-21 - do systemu LX 2500 /LOGABAX/
- MV 1664-RO - terminal odbiorczy:
- pojemność ekranu 1024 znaki /w 16 wierszach po 64 znaki/,
 - kineskop M31-310GH,
 - obudowa "ZEKOM R",
 - interfejs szeregowy RS 232C /V.24/ z obsługą linii 104 do 109; /lub pętla prądowa 20/60/mA,
 - szybkość transmisji nastawiana od 150 do 9600 bodów /albo 110 bodów/,
 - znacznik nieadresowany porusza się sekwencyjnie w najniższym wierszu,
 - automatyczne przeniesienie zapisu do następnego wiersza po przekroczeniu pojemności danego wiersza,
 - ilość obsługiwanych znaków jak w monitorze MV 1664; opcjonalnie można wybrać inną obsługę znaku HT: po odebraniu kodu HT następny znak interpretuje jako liczbę binarną /modulo 64/ ustalającą położenie znacznika w wierszu,
 - możliwość podłączenia drukarki DZM 180 współbieżnie z wyświetlaniem,

- wyposażenie dodatkowe:

- sprzęg SV 24/400 umożliwiające podłączenie do systemu MERA 400,
- uchwyt UM-22 umożliwiający zmianę ustawienia,
- podstawa UP-2,
- przystawka funkcyjna UF-2 do sterowania stronicowaniem,

MV 1664-KSR - terminal nadawczo - odbiorczy:

- interfejs szeregowy wg standardu RS-232C /V.24/ z pełną obsługą linii od nr 101 do 109 i/lub pętla prądowa 20V/60mA,
- klawiatura w układzie standardowym QWERTY z częścią numeryczną i klawiszami ustalającymi reżim współpracy monitora z komputerem:
 - z przełącznikami kontaktowymi - typ monitora MV 1664-KSR/EK,
 - z przełącznikami hallotronowymi - typ monitora MV 1664-KSR/EH,
- pozostała charakterystyka monitora jak dla MV 1664-RO.

SV 24/400 - pakiet sprzęgu umożliwiający podłączenie terminali posiadających interfejs RS-232C /V.24/ do minikomputera MERA 400:

- funkcjonalnie spełnia te same zadania, co jednostka sterująca SM-UZDAT 11 w systemie MERA 400 oraz dodatkowo obsługuje następujące przerwania operatora:
 - generowane przez naciśnięcie klawisza BREAK. Pakiet SV 24/400 wykrywa znak "BREAK" jako znak bez bitu stopu, generując sygnał błędny FE,
 - generowane dowolnie wybranym klawiszem, którego kod znaku jest ustawiony mikroprzełącznikami znajdującymi się na pakiecie, przerwanie to można uaktywnić lub nie mikroprzełącznikiem,
- wymiary i konstrukcja pakietu odpowiada funkcjonalnie jednostce sterującej SM-UZDAT 11,
- pakiet sprzęgu wyposażamy jedynie w złącza /bez kabla/ umożliwiające podłączenie go do terminala,

KT 1664/H - klawiatura alfanumeryczna z przełącznikami hallotronowymi /import/

- układ klawiszy typu QWERTY,
- blok alfanumeryczny, blok numeryczny oraz blok klawiszy funkcyjnych ON LINE, FULL DUPLEX, PAGING, NEXT /dwa ostatnie do sterowania wyświetlaniem w trybie stronicowania/
- klawisze SHIFT i CTRL - nie wpływają na funkcje klawiszy w bloku numerycznym,

- SHIFT: - niewciśnięty - duże litery i znaki z dolnego rejestru,
- wciśnięty - małe litery i znaki z górnego rejestru,
CTRL: znaki specjalne
- złącze 871 025 wraz z kablem dł.1m
 - wyjście w kodzie ASCII, negacje, 7 bitów danych + 1 sygnał sterujący + sygnały z klawiszy funkcyjnych,
 - zasilanie +5V/0,6A,
 - obudowa "ZEKOM RK"
- KA 1664/K - klawiatura alfanumeryczna z przekaźnikami kontaktronowymi prod.DOLAM:
- układ klawiszy typu QWERTY,
 - blok alfanumeryczny oraz blok klawiszy funkcyjnych ON LINE, FULL DUPLEX, F /zamiast F może być FAGING i dodatkowo NEXT, oba do sterowania wyświetlaniem w trybie stronicowania/,
 - repetycja - o częstotliwości 50Hz po upływie 1s od wciśnięcia i przytrzymania klawisza,
 - klawisze CAPS, SHIFT i CTRL:
 - CTRL - nadrzędny - znaki specjalne
 - CAPS niewciśnięty
 - SHIFT - niewciśnięty - małe litery, dolny rejestr,
 - wciśnięty - duże litery, górny rejestr,
 - CAPS - wciśnięty
 - SHIFT - niewciśnięty - duże litery, dolny rejestr,
 - wciśnięty - duże litery, górny rejestr
 - złącze 871 025 wraz z kablem dł.1m,
 - wyjście w kodzie ASCII, negacje, 7 bitów danych + 1 sygnał sterujący + sygnały z klawiszy funkcyjnych,
 - zasilanie +5V/0,5A
 - obudowa "ZEKOM RK"
- MV 2580 - terminal nadawczo-odbiorczy:
- pojemność ekranu - 2000 znaków na 1 stronie - 25 wierszy po 80 znaków
 - lub - 1920 znaków na 2 stronach oddzielonych pionową linią po 24 wiersze i 40 znaków,
 - zmianę formatu ekranu dokonuje się sekwencją znaków ESC, \backslash N oraz ESC, \backslash b
 - w pełni adresowalny znacznik,

- hardware oparty na mikroprocesorze MOTOROLA MC 6800,
- kineskop M31-310GH /12", luminofor P31/,
- obudowa: - części monitorowej - VELA 203 na podstawie UM21 prod. ZEKOM,
 - klawiatury - "ZEKOM RK"
- kompatybilny z terminalami VT 52 firmy DEC lub MERA 7953

- zestaw znaków - ASCII oraz znaki semigraficzne /które można ustalić na żądanie użytkownika/,
- kopiowanie ekranu /tryb COPY SCREEN/ lub współbieżnie z wyświetlaniem /tryb AUTO COPY/,
- strohicowanie /tryb HOLD SCREEN/ z wyborem podsyłania jednej linii lub całego ekranu,
- możliwy zapis znaku w trybie INVERSE,
- przy pracy ON LINE możliwa jest transmisja w dupleksie albo w półdupleksie; możliwa jest również praca off-line,
- interfejs szeregowy: napięciowy RS 232C /CCTT V.24/ z obsługą wszystkich linii od 101 do 109 lub prądowy typu "current loop" 20/60mA; szybkość nastawiana od 75 do 9600 bodów,
- interfejs równoległy do podłączenia drukarki DZM 180 /9100/
- funkcje sterujące terminala podzielono na trzy grupy:
 1. grupa CTRL - 17 znaków
 - NUL - brak reakcji
 - STX, ETX - dla transmisji blokowej
 - BS - cofnięcie znacznika
 - HT, VT - tabulacje pozioma i pionowa
 - LF, CR - wysuw linii, powrót znacznika
 - FF - zerowanie ekranu
 - DC1, DC3 - żądanie wznowienia lub wstrzymania transmisji
 - ESC - wywołanie sekwencji sterującej
 - SJ, SO - obsługa trybu INVERSE
 - BEL - sygnał akustyczny
 - SUB - zapowiedź następnego znaku jako semigraficzny
 - US - umieszczenie pustego znaku na ekranie
 2. grupa ESC - 20 sekwencji znaków
 - ESC,A; ESC,B; ESC,C; ESC,D - jednokrotne sterowanie ruchem znacznika

- ESC, Y, "R", "C" - bezpośrednie sterowanie ruchem znacznika do określonego punktu,
- ESC, F; ESC, G - obsługa trybu SEMIGRAPHIC
- ESC, J - ruch znacznika w górę z rolowaniem obrazu w dół dla najwyższego wiersza
- ESC, H - ustawienie znacznika w lewym górnym rogu ekranu
- ESC, J; ESC, K - kasowanie znaków za znacznikiem odpowiednio do końca ekranu, i wiersza
- ESC, S - zmiana formatu ekranu
- ESC, T - wyświetlenie aktualnych nastaw parametrów transmisji
- ESC, X - wysłanie do komputera kodu znaku w miejsce znacznika
- ESC, Z - monitor wysła informacje o trybie swojej pracy
- ESC, [; ESC, \ - obsługa trybu HOLD SCREEN
- ESC,] ; - obsługa trybu COPY SCREEN
- ESC, ^ ; ESC, - - obsługa trybu AUTO COPY

3. grupa F - 12 kombinacji - służy do lokalnego /off-line/ sterowanie funkcjami monitora za pomocą dodatkowego klawisza F

- F, 2; F, 4; F, 5; F, 6 - jednokrotne sterowanie ruchem znacznika
- F, C; F, A - obsługa trybu COPY SCREEN
- F, V; F, O - obsługa trybu AUTO COPY
- F, [; F, \ - obsługa trybu HOLD SCREEN
- F, I; F, N - zezwolenie na wprowadzenie odpowiednio następną linię, następnego ekranu.

- wyposażenie dodatkowe

- pakiet sprzęgu SV 24/400 umożliwiający podłączenie do systemu MERA 400,
- uchwyt UM-21 do zamocowania jako urządzenie wolnostojące,
- uchwyt UM-1 do zamocowania na drukarce DZM 180 w wersji MV 2580A /jako monitor ekranowy współpracujący z drukarką/

Gotowa jest do wprowadzenia do produkcji o ile znajdzie zainteresowanie u użytkowników rodzima tanich terminali ekranowych ME 1664, o następującej charakterystyce:

- ME 1664 - KSR - terminal nadawczo - odbiorczy
- ME 1664 - RO - terminal odbiorczy

- interfejs szeregowy wg standardu RS 232C /CCITT V.24/
obsługa wyłącznie linii TxD /tylko wersja KSR/ i RxD:
lub pętla prądowa 20 mA

- szybkość transmisji 150 lub 300 bodów
- realizuje następujące funkcje specjalne:
 - LF - wysuw linii i powrót znacznika do początku linii /nie realizuje odrębnej funkcji CR/,
 - BS, FR, BEL, NUL,
- ruch znacznika w górę po odebraniu znaku CTRL, p,
- obudowa telewizora VELA 203,
- niska cena,
- wyposażenie dodatkowe:
 - pakiet sprzętu Sv 24/400 umożliwiający niezależne podłączenie do systemu MERA 400,
 - uchwyt UM-21 lub podstawa UP-1 w przypadku użycia monitora jako urządzenia wolnostojącego, umożliwiające zmianę ustawienia względem operatora,
 - przystawka funkcyjna UP-12 do sterowania stronicowaniem

MV 1664 - monitor ekranowy odbioreczy

- interfejs równoległy o strukturze analogicznej jak w interfejsie drukarki DZM 180,
- kod ASCII /2 zestawy po 64 znaki/,
- szybkość transmisji 60 znaków/s,
- obudowa telewizora VELA 203,
- typowe zastosowanie: współpraca z drukarką DZM 180,
- niska cena,
- wyposażenie dodatkowe: jak w przypadku monitora MV 1664.

Oferta nasza obejmuje również instalacje produkowanych przez nas urządzeń, nawet jeśli są nietypowe; wtedy wymagane jest uzgodnienie warunków.

Mgr inż. Jerzy NAGÓRSKI
Zakład Elektroniki Komputerowej
ul. Makowa 8, 91-480 Łódź
tel. 34-30-49

SPÓŁKA USŁUG INFORMATYCZNYCH "INFORMATYKA"
02-632 WARSZAWA, ul. Dereniowa 7 m 26

F H S - nowy Job-Control SOM3

cena 120' tys. zł.

Umożliwia korzystanie z biblioteki plików /zbiorów/ na dowolnej sekcji dyskowej. Pliki biblioteczne identyfikowane nazwą z rozszerzeniami mogą być pobierane z biblioteki na dowolną roboczą sekcję z maksymalną dla Mery 400 prędkością. Wpis pliku do biblioteki odbywa się również przez szybkie kopiowanie. W trakcie użytkowania biblioteki nie jest potrzebna kompresja dysku.

FHS jest procesorem pełniącym rolę job-controla, umożliwia więc wykonywanie wszystkich operacji na strumieniach, ładowanie i wykonywanie innych procesorów i programów oraz zawiera mechanizm makroinstrukcji /makrodyrektyw, procedur/. Ładowanie innego programu w obszar użytkownika może odbyć się z biblioteki SOM, z biblioteki FHS lub z dowolnego strumienia.

FHS będąc organizatorem biblioteki plików wykonuje na bibliotece operacje pobrania, wpisania, skasowania pliku, zmiany nazwy. Seria dodatkowych programów - procesorów umożliwia wydrukowanie listy zakatalogowanych plików, zrzucenie katalogu na zapasowy obszar dysku - zabezpieczenie przed wymazaniem. Istnieje też mechanizm odtwarzania uszkodzonych bibliotek.

M A X - nowy makroassembler

cena 80 tys. zł.

MAX jest jednonakładowym makroassemblerem wymagającym od 12k pamięci przewidzianym na szybką kompilację programów napisanych pierwotnie w Fortranie. Pracuje tylko w standardzie GCW. Zawiera rozszerzony mechanizm makrodefinicji pozwalający na definiowanie ich np. we wstawkach assemblerowych w Fortranie. Sam stawia znacznik EOF na strumieniu B0, koniec kompilacji segmentu potwierdza na lokalnym CO. Dobra diagnostyka błędów, nie rozwlekły listing, tablica symboli drukowana w kolejności alfabetycznej.

Oprogramowanie powyższe pracuje w standardowym systemie SOM, jedynie FHS wymaga niewielkich zmian w systemie możliwych do przeprowadzenia bez generacji systemu operacyjnego.

Zamówienia można kierować na adres Spółki.

23 października 85r.

Zakład Elektroniczny
Mieszko Ciepłiński
ul. Na Uboczu 18
02-791 Warszawa

Do Szanownych Użytkowników
Systemów Komputerowych
MERA 400

Z prawdziwą przyjemnością pragniemy poinformować Państwa, że zakład nasz oferuje wykonanie, wraz z instalacją, modułów pamięci dodatkowej do systemów MERA 400.

Oto podstawowe dane urządzenia:

- pojemność 256k słów /17 bitów/,
- oparta na półprzewodnikowych układach pamięci dynamicznych produkcji zachodniej,
- możliwość rozszerzenia /w ramach tego samego zestawu pakietów/ do pojemności 512k,
- zawarta na 3 pakietach,
- umożliwia operowanie blokami pamięci o poj. 4k ,
- bezkonkurencyjnie niska cena - 1210 tys.zł - obejmuje instalację i roczną gwarancję.

Ewentualne zamówienia prosimy kierować do:

Rzemieślniczej Spółdzielni w Piasecznie
05-500 Piaseczno, ul. Warszawska 1
Zakład M. Ciepłiński

Z poważaniem

M. Ciepłiński

Zakład Systemów Cyfrowych "DIGITEX"
81-832 Sopot, ul. Mickiewicza 15
inż. K. Wejs, tel. 51-09-53

TERMINAL KOMPUTEROWY "DIGITEX"

Terminal może współpracować z każdym komputerem posiadającym interfejs szeregowy.

Terminal jest zbudowany w oparciu o mikroprocesor i układy peryferyjne wielkiej skali integracji co zapewnia dużą niezawodność i pozwala na dostosowywanie jego parametrów do indywidualnych wymagań użytkowników.

Parametry techniczne /wersja podstawowa/:

- ekran 12" o zielonej poświacie
- 25 linii x 80 kolumn /32 linie x 64 kolumny/
- zestaw znaków: 96 ASCII + 96 graficznych
- sterowanie kursorem w obrębie całego ekranu, zerowanie ekranu/linii, usuwanie linii/znaku, wstawianie linii/znaku
- interfejs V24 i pętla prądowa
- prędkość transmisji 110 do 19200 bodów

Na życzenie zamawiającego istnieje możliwość dostarczenia terminali:

- ze specjalnym zestawem wyświetlanych znaków
- ze specjalnym układem klawiatury i klawiszy funkcyjnych
- z dodatkowym interfejsem szeregowym /asynchronicznym lub synchronicznym/
- z rozszerzoną pamięcią ekranu do 4 stron
- ze specjalnym protokołem transmisji
- z interfejsem równoległym /np. do drukarki/

Zamówienia proszę kierować na adres: Spółdzielnia Rzemieślnicza
"BUDOWA" 81-364 Gdynia ul. 10 Lutego 33 tel. 20-55-11.

TOBI - PRZEDSIĘBIORSTWO ZAGRANICZNE W POLSCE
05-120 LEGIONOWO, ul. Jagiellońska 8
tel. 74-27-64

RS 232C/V.24 F I B E R O P T I C M O D E M

The RS 232C/V.24 Fiber Optic Modem is a complete ready to operate optical fiber transmission system designed to transmit digital data through modulated light over distances of up to 500 meters. The system consists of an optical transmitter, receiver and signal conversion module that is fully compatible with RS 232C/V.24 standard. The system is suitable for simplex or full duplex asynchronous data transmission at up to 76,8 kbauds. A built in 220 V AC power supply furnishes all operating powers and an EIA 25 - pins connector allows the system to be quickly "plugged - in" to existing equipment in a matter of moments.

F E A T U R E S :

- full duplex, dual channel modem;
- secure data transmission;
- low bit error rate /less than 10^{-9} /;
- transmission EMI/RFI immunity;
- electrical isolation;
- arbitrary data format;
- long distance operation;
- low installation cost due to light weight fiber optic cable.

T E C H N I C A L S P E C I F I C A T I O N :

- data rate 0 - 76,8 kbps;
- input/output levels RS 232C/V.24;
- transmission span 500 m.;
- power consumption 10VA;
- dimensions 230 x 150 x 50;
- weight ca. 1kg.

P R I C E :

A complete System with	- 20 m. F.O. Cable	- 400.000,- zł
/Short distance version/	- 50 m. "	- 450.000,- zł
	-100 m. "	- 500.000,- zł
A complete System with	-200 m. "	- 700.000,- zł
/Long distance version/	-300 m. "	- 800.000,- zł
	-400 m. "	- 900.000,- zł
	-500 m. "	-1000.000,- zł

M O R E I N F O R M A T I O N S :

mgr inż. Jacek MORAWSKI, Warszawa tel. 25-54-03
Tuesday and Thursday - 9⁰⁰ - 12⁰⁰.

LISTA UCZESTNIKÓW
KONFERENCJI UŻYTKOWNIKÓW MINIKOMPUTERA MERA-400

1. ADAMCZYK Zbigniew, Zakład Informatyki Spółdzielni Mieszkaniowych, ul. Kliny 2, Kraków, tel.sł. 11-94-67. 31-465 Kraków.
2. ANTOSIK Tomasz, PP Polmozbyt, ul. Strykowska 1/5, 91-725 Łódź, tel.sł. 84-92-42.
3. BADURA Wojciech, Huta Szkła Okiennego "KARA", ul. Paplińskiego 1, 97-300 Piotrków Tryb., tel.sł. 16-11-41 wew. 97.
4. BARCZYK Stanisław, Huta Szkła Walcowanego, ul. Kolejowa 1, /32-312 Jaroszewiec, tel.sł. 309-14 wew. 143.
5. BAUER Zbigniew, Zakład Informatyki Spółdzielni Mieszkaniowych, ul. Kliny 2, 31-485 Kraków, tel.sł. 11-94-67.
6. BESTYŃSKI Piotr, Poznańskie Biuro Projektów Budownictwa Przemysłowego, ul. Ratajczaka 10/12, Poznań, tel.sł. 622-31 wew. 151. 60-567 Poznań.
7. BICZEL Bartłomiej, Centralny Ośrodek Metodyczny Poradnictwa Wychowawczo-Zawodowego Min.Oświaty i Wychowania, ul. Nowowiejska 5, 00-643 Warszawa, tel.sł.25-63-59.
8. BIEGAŁA Ludwik, Instytut Niskich Temperatur i Badań Strukturalnych PAN, Pl. Katedralny 1, 50-950 Wrocław, tel.sł. 22-10-71 wew. 72.
9. BIEŃKOWSKI Andrzej, Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne, ul. Zwycięstwa 140, 75-613 Koszalin, tel. sł. 277-51 wew. 121.
10. BIERNAT Mirosław, Politechnika Wrocławska Instytut Chemii Org. i Fizycznej, ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, tel.sł. 20-37-83.
11. BIESIADOWSKI Jerzy, G.Z.E. UNIMOR, ul. Rzeźnicka 54/56, 80-822 Gdańsk, tel.sł. 375-330.
12. BOBCOW Andrzej, Dział Planowania Informatyki Stoczni Remontowej "RADUNIA", ul. Na Ostrowiu 1, 80-873 Gdańsk, tel.sł, 31-68-31 wew. 278.
13. BONIECKI Marek, Ośrodek Informatyki Centralnego Związku Spółdzielczości Pracy, ul. Bonifaterska 14, 00-213 Warszawa, tel.sł. 31-41-63, 31-68-72.
14. BOROWSKI Eugeniusz, Zakłady Metali Lekkich "Kęty", ul. Kościuszki 111, 32-650 Kęty, tel.sł. 522-51, wewn. 217.
15. BRANIECKI Andrzej, Instytut Okrętowy Politechniki Gdańskiej, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł.47-16-43.

16. BRAUN Mirosław, PP Polmozbyt, ul. Strykowska 1/5, 91-725 Łódź, tel.sł. 84-92-42.
17. BRZESKI Michał, Ośrodek Elektronicznego Przetwarzania Danych, ul. Krasinskiego 16, 01-581 Warszawa, tel.sł. 39-94-54.
18. BUJAK Andrzej, Cieczynska Fabryka Farb i Lakierów "Polifarb", 43-400 Cieszyn Marklowice, tel.sł. 214-10.
19. CIAS Andrzej, Międzyresortowy Instytut Przemysłu Budowlanego, Politechnika Warszawska, ul. Armii Ludowej 16, 00-637 Warszawa, tel.sł. 210361 wewn. 6587.
20. CICHOCKI Mieczysław, Biuro Projektowo-Badawcze Budownictwa Ogólnego Miastoprojekt, Łódź, ul. Traugutta 21, tel.sł. 338119 lub 328920 wewn. 248137. 90-950 Łódź.
21. CICHOWSKA Gracjana, Instytut Chemii Fizycznej PAN, ul. Kasprzaka 44/52, 01-224 Warszawa, tel.sł. 32-32-21 wewn. 331.
22. CZAJKOWSKI Piotr, Biuro Projektów Budownictwa Komunalnego, ul. Grunwaldzka 2, 82-300 Elbląg, tel.sł. 259-08.
23. CZERNIAK Zbigniew, Instytut Okrętowy Politechniki Gdańskiej, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł.47-17-08.
24. CZERWIŃSKI Michał, OBR Elektronicznej Aparatury Medycznej, ul. Wolności 345a, 41-800 Zabrze, tel.sł. 71-64-21 - 9, wewn. 255.
25. CZERWOSZ Leszek, Centrum Medycyny Doświadczalnej i Klinicznej PAN - ul. Dworkowa 3, 00-784 Warszawa, tel.49-74-88, 49-66-51.
26. DASZCZUK Wiktor, Instytut Melioracji Użytków Zielonych Falenty, 05-550 Raszyn, tel. 500-531, wewn. 243.
27. DOBROWOLSKI Marek, Biuro Projektów Budownictwa Wiejskiego, ul. J.Nowickiego 30/32, 87-100 Toruń, tel.wewn 58, tel.sł. 210-51.
28. DOKIMUK Jerzy, Politechnika Łódzka, ul. Żwirki 36, Łódź, 90-924, tel.sł. 365522 wewn. 938.
29. DRÓZDZ Jolanta, Politechnika Łódzka, ul. Żwirki 36, 90-924 Łódź, tel.sł. 365522 wewn. 938.
30. DUSZAŃSKI Marek, Śląski Mostostal M4, ul. Wolności 191, 41-800 Zabrze, tel.sł. 71 32 21 wewn. 202.
31. DZIADURA Stanisław, Ośrodek Badawczo-Rozwojowy Metrologii Elektrycznej, ul. Sulechowska 1, 65-022 Zielona Góra, tel.wł. 630-56, wewn, 5371.
32. DŻOGA Jerzy, Przedsiębiorstwo Zagraniczne "AMEPOL", ul. Plac Żelaznej Bramy 1, 00-136 Warszawa, tel.sł. 27-31-12.

33. DYMEK Anna, Krakowskie Biuro Projektowo-Badawcze, Budown.-
Ogólnego. Al. Powstańców Warszawy 10, 31-549 Kraków,
tel.sł. 11-87-33 w. 347.
34. DYRGA Tadeusz, Instytut Elektroenergetyki i Automatyki,
Politechniki Gdańskiej, ul. Majakowskiego 11/12,
tel.sł. 47-24-39.
35. FERET Błażej, Międzyrasortowy Instytut Techniki Radiacyjnej
Politechniki Łódzkiej, ul. Wróblewskiego 15, 93-590
Łódź, tel.sł. 36-55-22 wewn. 12-33.
36. FIGIEL Zbigniew, Centrala Narodowego Banku Polskiego, ul.
Świętokrzyska 11/21, 00-950 Warszawa, tel.sł.20-03-31
wewn. 22-41.
37. FINDEISEN Piotr, Instytut Informatyki Uniwersytetu Warszawskie-
go PKiN, 00-901 Warszawa, tel.sł. 200-211, wewn. 2103.
38. GALĄZKA Stanisław, Centrum Badawczo-Wdrożeniowe "MERCOMP",
ul. Poezji 19, 04-994 Warszawa, tel.sł. 25-56-40.
39. GAWĘCKI Michał, Biuro Projektów Budownictwa Wiejskiego,
ul. Piekary 17, Poznań, tel.sł. 33-05-81. wewn. 570.
40. GLURA Wiesław, Ogólnouczelniany Ośrodek Obliczeniowy UMK,
ul. Gagarina 7, 87-100 Toruń, tel.sł. 260-17 wewn.44.
41. GOŁĘBIEWSKA-WALCZAK Krystyna, Politechnika Łódzka, Instytut
Aparatów Elektrycznych, ul. Żwirki 36, Łódź, tel.
sł. 36 55 22 wewn. 683a 80-924 Łódź.
42. GÓRSKI Krzysztof, Ośrodek Obliczeniowy WZSP, ul. Dyrekcyjna 5,
80-852 Gdańsk, tel.sł. 20-82-15 i 31-21-83.
43. GRALL Jerzy, Instytut Włókiennictwa, ul. Gdańska 91/93,
90-950 Łódź, tel.sł. 33-96-00, dom. 32-76-54.
44. GRALL Tadeusz, COBR Maszyn Włókienniczych "POLMATEX CENARO",
ul. Wólczańska 55/59, 90-608 Łódź, tel.sł. 32-85-70,
wewn. 243, dom. 78-50-16.
45. GRZYBOWSKI Ryszard, Biuro Studiów i Projektów Energetycznych
Energoprojekt, ul. Piekary 19, 60-067 Poznań, tel.
sł. 222-011 wewn. 298.
46. GUTKOWSKA Danuta, Warszawskie Przedsiębiorstwo Konstrukcji
Stalowych i Urządzeń Przemysłowych "MOSTOSTAL",
ul. Bracka 4, Warszawa, tel.sł. 26-1231, wewn. 208.
47. HALSKI Adam, Śląski Mostostal M4, ul. Wolności 191, 41-800
Zabrze, tel.sł. 71-32-21 wewn. 205.
48. HANKE Roman, Żyrardowskie Zakłady Tkanin Technicznych,
ul. Okrzei 51, 96-300 Żyrardów, tel.sł. 20-31/5 w.264.

49. HORAK Andrzej, Śląska Akademia Medyczna, ul. Poniatowskiego 15, Katowice, tel. śl. 527095.
50. JABŁOŃSKA Gabriela, Główne Biuro Studiów i Projektów Przeróbki Węgla - SEPARATOR, ul. Armii Czerwonej 2, 40-952 Katowice. tel. śl. 586-071, wewn. 513, lub 473.
51. JABŁOŃSKI Włodzimierz, Biuro Projektów Budownictwa Elektrowni i Przemysłu "ELPRO", ul. Jasna 14/16, 00-950 Warszawa, tel.śl. 27-50-81.
52. JEZIERSKA-ZIEMKIEWICZ Elżbieta, Przedsiębiorstwo Zagraniczne "AMEPOL", ul. Plac Żelaznej Bramy 1, 00-136 Warszawa, tel.są.
53. KALIŃSKI Adam, Ośrodek Informatyki Centralnego Związku Spółdzielczości Pracy, ul. Bonifaterska 14, Warszawa, tel.śl. 31-41-63 lub 31-68-72.
54. KAPALA Zenon, Instytut Okrętowy Politechniki Gdańskiej, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.śl. 47-21-32.
55. KAPCIA Jerzy, Instytut Telekomunikacji Politechniki Gdańskiej, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.śl. 47-18-08.
56. KAPCZYŃSKI Roman, Instytut Inżynierii Lądowej Politechniki Poznańskiej, ul. Pl. Curie Skłodowskiej 5, 60-965 Poznań, tel.śl. 782-418.
57. KARCZEŃSKI Włodzimierz, Biuro Projektowo Badawcze Budownictwa Ogólnego MIASTOPROJEKT 2, ul. Więckowskiego 20, 90-722 Łódź, tel. śl. 328100 wewn., 35.
58. KAREWICZ Bogusław, Zakład Rzemieślniczy, ul. Makowa 8, 91-480 Łódź,
59. KASPRZAK Andrzej, Instytut Obróbki Skrawaniem, ul. Wrocławska 37a, 30-011 Kraków, tel. śl. 33-93-33 wewn. 219.
60. KAZIMIERCZAK Bogdan, Instytut Elektroenergetyki i Automatyki Politechniki Gdańskiej, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.śl. 47-23-39.
61. KĘDZIERSKI Stanisław, Śląski MOSTROSTAL M4, ul. Wolności 191, 41-800 Zabrze, tel.śl. 71-32-21 wewn. 202.
62. KOBZA Marian, Instytut Techniki Ciepłej, ul. Dąbrowskiego 113, 90-950 Łódź, tel.śl. 43-26-50.
63. KOCHEL Tadeusz, Górnośląski Okręgowy Zakład Gazownictwa, ul. Gwardii Ludowej 11, 41-800 Zabrze, tel.śl. 71-52-21 wewn. 5415, tel.domowy 71-21-66.
64. KONECKA Ewa, Dział Informatyki Wojewódzkiego Przedsiębiorstwa Energetyki Ciepłej, ul. Dzierżyńskiego 5, 85-315 Bydgoszcz, tel.śl. 342-81 wewn. 288.

65. KONIECZNY Roman, Instytut Transportu Politechniki Śląskiej,
ul. Krasińskiego 8, 40-012 Katowice, tel.śł. 513-677.
66. KORYTEK Andrzej, Ośrodek Badawczo-Rozwojowy Systemów Automatyki,
ul. Armii Czerwonej 66/72, 61-807 Poznań, tel.śł.
456-81 lub 456-82 wewn. 4.
67. KOSECKI Mieczysław, Biuro Projektów Budownictwa Morskiego
"BIMOR", ul. Pl. Batorego 4, 70-207 Szczecin, tel.śł.
403-305, tel.dom. 419-71.
68. KOWALCZYK Aleksandra, Warszawskie Przedsiębiorstwo Konstrukcji
Stalowych i Urządzeń Przemysłowych, ul. Bracka 4,
Warszawa, tel.śł. 26-12-31 wewn. 208.
69. KOWALCZYK Bogdan, Instytut Technologii Mechanicznej Politechniki
Warszawskiej, ul. Marabutta 86, 02-524 Warszawa, *
tel.śł. 49-98-71 wewn. 206.
70. KOWALCZYK Władysław, Biuro Projektów Budownictwa Morskiego
"BIMOR", Pl. Batorego 4, 70-207 Szczecin, tel.śł. 403305.
71. KOWALSKI Stanisław, Centralny Ośrodek Informatyki Dragownictwa,
ul. Pretfina 9/11, Wrocław, tel.śł. 61-59-70.
72. KRAJEWSKA Maria, Biuro Projektów NAFTOPROJEKT, ul. Mysia 3,
00-498 Warszawa, tel.śł. 28-40-21 wewn. 88 lub 32,
tel.dom. 42-45-46.
73. KRALKA Wiktor, Zakłady Metali Lekkich "Kęty", ul. Kościuszki 111,
32-650 Kęty, tel.śł. 522-51, wewn. 154.
74. KRUPA Jacek, Zakład Usług Technicznych ZWKTiR, ul. Bartycka 20,
00-716 Warszawa, tel.śł. 40 00 21 wewn. 212.
75. KRAWCZYK Zdzisław, CBS i PBW HYDROPROJEKT, ul. Powstańców
Warszawskich 202, O/Gdańsk, tel.śł. 41-20-61.
76. KRYCH Jacek, WZT TELKOM 7 TELETRA, ul. Bułgarska 67/73, 60-320
Poznań, tel.śł. 676 810, wewn. 404 lub 412.
77. KRYGOWSKI Józef, Biuro Projektów Budownictwa Wiejskiego,
ul. M. Fornalskiej 1, 35-959 Rzeszów.
78. KUBRYCHT Jerzy, Ośrodek Badawczo Rozwojowy Elektronicznych Ukła-
dów Specjalizowanych, ul. Grudziądzka 46, 87-101 Toruń,
tel.śł. 330 45 wewn. 401.
79. KRZEMIEN Aleksander, Kopalnia Węgla Kamiennego Śiersza", w.
Trzebini, woj. Katowice, Trzebina 9 wewn. 5251.
Krzeszowice 206-13.
80. KULESZA Joanna, Centrum Medycyny Doświadczalnej i Klinicznej
PAN, ul. Dworkowa 3, 00-784 Warszawa, tel.śł. 49-74-88,
49-66-51.

81. KWAPIŃSKI Henryk, Zakłady Automatyki Chemicznej "Metalchem",
ul. Chorzowska 44 b, Gliwice, tel.sł. 31-64-41 wewn.102.
44-101 Gliwice.
82. KWAŚNICKA Hanna, OBR Elektronicznej Aparatury Medycznej, ul.
Wolności 345 a, 41-800 Zabrze, tel.sł. 71-64-21 - 9,
wewn. 255.
83. LASOTA Jarosław,
84. LEWANDOWSKI Marek, Przedsiębiorstwo Zagraniczne "AMEPOL",
Plac Żelaznej Bramy 1, 00-136 Warszawa, tel.sł. 20-34-75.
85. LEWIŃSKI Roman, Centrum Badawczo-Wdrożeniowe "MERCOMP",
ul. Poezji 19, 04-994 Warszawa, tel.sł. 25-56-40.
86. LITWINIUK Andrzej, Instytut Informatyki, Uniwersytetu Warszawskie-
go, Warszawa, tel.sł. 200-211 wewn. 2394.
87. ŁOZIŃSKA Grażyna, Międzynarodowy Instytutu Przemysłu Budowlanego,
ul. Armii Ludowej 16, 00-637 Warszawa, tel. sł.25-78-56,
tel.dom. 27-93-00.
88. MACHOWIAK Bogdan, Biuro Projektów Budownictwa Wiejskiego,
ul. Piekary 17, 60-959 Poznań, tel.sł. 33-05-81, wewn.569.
89. MACIUK Bronisław, Politechnika Śląska, ul. Krasińskiego 8,
40-019 Katowice, tel.sł. 516671, wewn. 24.
90. MALINOWSKI Marek, Politechnika Warszawska, ul. Łukasiewicza 17,
Płock, tel.sł. 260-61, wewn. 227.
91. MAŁOWANA Wiesława, PPPW PREFBET Rybnicki Zakład Prefabrykacji ,
ul. Wiejska 7, Rybnik, tel.sł. 26451.
92. MARTIN Aleksander, Huta Miedzi "GŁOGÓW", 67-231 Żukowice,
tel.sł. 320-71, wewn. 6380.
93. MARTIN Włodzimierz, Instytut Okrętowy Politechniki Gdańskiej,
ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł.47-21-32.
94. MACZYŃSKI Sławomir, Instytut Melioracji i Użytków Zielonych,
Falenty, 05-550 Raszyn, tel.sł. 500-531, wewn. 243.
95. MELLER-KAPCIA Maria, Instytut Okrętowy Politechniki Gdańskiej,
ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł.47-21-32.
96. MICHAŁEC Janusz, Zespół Opieki Zdrowotnej, ul. Kamieniec 10,
Zakopane, tel.sł. 20-21, wewn. 288.
97. MICHAŁOWSKI Henryk, Instytut Melioracji i Użytków Zielonych
Falenty, 05-550 Raszyn, tel.sł. 500-531, wewn. 243.
98. MIERZEJEWSKI Adam, CTH PP "Prema", ul. Krakowskie Przedmieście
47/51, 00-950 Warszawa, tel.sł. 26-32-01, wewn. 344.
99. MIKULSKA Maria, Politechnika Łódzka, ul. Żwirki 36, 90-924 Łódź,
tel.sł. 36-55-22, wewn. 262.

100. LORAWSKA Barbara, P.Z. "TOBI", ul. Jagiellońska 8, Legionowo, tel.sł. 25-54-03.
101. MORAWSKI Jacek, Przedsiębiorstwo Zagraniczne w Polsce "TOBI", ul. Jagiellońska 8, Legionowo, tel.sł. 25-54-03.
102. NAGÓRSKI Jerzy, Zakład Elektroniki Komputerowej, ul. Makowa 8, 91-490 Łódź.
103. NAZAROWSKI Jerzy, Ośrodek Postępu Technicznego NOT, ul. Bartycka 20, Warszawa, tel.sł. 22-23-31.
104. NIEMIEC Józef, Ośrodek Postępu Technicznego NOT, ul. Bartycka 20, Warszawa, tel.sł. 22-77-23, 22-23-31.
105. NIKIEL Wojciech, Przemysłowy Instytut Automatyki i Pomiarów MERA-PIAP, Al. Jerozolimskie 202, 02-222 Warszawa, tel.sł. 23-70-71 wewn. 489.
106. NIKODEMSKA Joanna, Pracownia Usług Obliczeń Ośrodka Obliczeniowego WZSP, ul. Dyrekcyjna 5, 80-852 Gdańsk, tel.sł. 31-56-21 wewn. 241.
107. NIKODEMSKI Marek, Instytut Okrętowy Politechniki Gdańskiej, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł. 47-18-08.
108. NIWIŃSKI Stanisław, Instytut Elektrotechniki, ul. Pożaryskiego 28, 04-703 Warszawa, tel.sł. 12-31-53.
109. NOWACKI Kazimierz, CBS i PBW HYDROPROJEKT, ul. Dubois 9, 00-182 Warszawa, tel.sł. 387041 wewn. 246.
110. NOWAK Dorota, Śląska Akademia Medyczna, ul. Poniatowskiego 15, Katowice, tel.sł. 527-095.
111. NOWAK Piotr, Politechnika Wrocławska Instytut Chemii Org. i Fizycznej, ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, tel.sł. 20-24-85.
112. OCZKI Zbigniew, OBR Elektronicznych Układów Specjalizowanych, ul. Grudziądzka 46, 87-100 Toruń, tel.sł. 33045 wewn. 303.
113. OLECH Bogusława, PSM Zakład Nr 7, ul. Pekin 1, 41-201 Sosnowiec, tel.sł. 699-144.
114. OLKOWSKI Henryk, Instytut Ciężkiej Syntezy Organicznej Błachownia, ul. Energetyków 9, 47-232 Kędzierzyn-Koźle, tel.sł. 332-41 wewn. 5459.
115. OLDZIEJ Danuta, Dział Informatyki Wojewódzkiego Przedsiębiorstwa Energetyki Ciepłej, ul. Dzierżyńskiego 5, 85-315 Bydgoszcz, tel.sł. 342-81 wewn. 288.
116. OWCZAREK Konstanty, MFK - Zakład Ekonomiki i Informatyki, ul. Piotrkowska 147, 90-440 Łódź, tel.sł. 366355.

117. PERCZAK Zdzisław, Pomorski Okręgowy Zakład Gazownictwa w Gdańsku, Zakład Gazowniczy Bydgoszcz, ul. Jagiellońska 42, 85-097 Bydgoszcz, tel.sł. 220081.
118. PEREK Maria, Huta Szkła Okiennego "Szczakowa", ul. Kolejarzy 81, 32-520 Jaworzno, tel.sł. 774-41 wewn. 120.
119. PILASIEWICZ Barbara, Politechnika Warszawska, ul. Armii Ludowej 16, 00-637 Warszawa, tel.sł. 25-75-40.
120. PISIEWICZ Andrzej, Wojewódzkie Biuro Projektów, ul. Wolności 286, 41-800 Zabrze, tel.sł. 71-20-21 wewn. 84.
121. PLACZEK Denuta, PPFW PREFBET Rybnicki Zakład Prefabrykacji ul. Wiejska 7, Rybnik, tel.sł. 26451.
122. PRZYSYŁEK Jan, Komenda Wojewódzka Straży Pożarnych, ul. Bema 17, 82-300 Elbląg, tel.sł. 276-91, wewn. 213.
123. PSIORZ Marek, Zarząd Portu Szczecin-Świnoujście, ul. Bytomska 7, Szczecin, tel.sł. 308-390.
124. RADOMSKA Maria, Politechnika Wrocławska Instytut Chemii Org. i Fizycznej, ul. ybrzeże Wyspiańskiego 27, 50-370 Wrocław, tel.sł. 20-24-12.
125. RADOMSKI Ryszard, Politechniki Wrocławska Instytut Chemii Org. i Fizycznej, ul. Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, tel.sł. 20-24-12.
126. REWILAK Zygmunt, Zakład Informatyki Spółdzielni Mieszkaniowych Spółdzielnia Osób Prawnych, ul. Klíny 2, Kraków, tel. sł. 11-94-67. 31-485 Kraków.
127. ROGALSKI Roman, Pomorski Okręgowy Zakład Gazownictwa w Gdańsku, Zakład Gazowniczy Bydgoszcz, ul. Jagiellońska 42, 85-097 Bydgoszcz, tel. sł. 22-00-81.
128. RÓŻYCKI Dobrosław, MPK Zakład Ekonomiki i Informatyki, ul. Piotrkowska 147, 90-440 Łódź, tel.sł. 366355.
129. RUDZIŃSKI Jan, Instytut Informatyki Uniwersytetu Warszawskiego, PKiN, 00-910 Warszawa, tel.sł. 200-211 wewn. 2103.
130. SALWICKI Andrzej, Instytut Informatyki UW, Warszawa, tel.wewn. 200211 wewn. 2394. PKiN, 00-901 Warszawa.
131. SITEK Stanisław, WZT Telekom-Telettra, ul. Bułgarska 67/73, 60-320 Poznań, tel.sł. 676801 wewn. 404 lub 412.
132. SMOL Anna, Zespół Opieki Zdrowotnej, ul. Kamieniec 10, Zakopane, tel.sł. 20-21 wewn. 288.
133. SOPEK Mirosław, Międzyresortowy Instytut Techniki Radiacyjnej Politechniki Łódzkiej, ul. Wróblewskiego 15, 93-590 Łódź, tel.sł. 36-55-22, wewn. 12-33.

134. STECURA Jadwiga, Przedsiębiorstwo Zagraniczne "AKMIPOL",
Plac Żelaznej Bramy 1, 00-136 Warszawa, tel.sł. 20-34-75.
135. STEFANIAK Tomasz, Biuro Projektów Przem. Lekkiego "BEDETE",
Pl. Zwycięstwa 2, 90-950 Łódź, tel.sł. 36-33-44 wewn.113.
136. SZANSER Wojciech, CBW MERCOP, ul. Poezji 19, 04-994 Warszawa,
tel.sł. 25-56-40.
137. SZCZEPANIAK Barbara, Biuro Projektów Budownictwa Komunalnego,
ul. Tuwima 22/26, 90-002 Łódź, tel.sł. 32-32-75.
138. SZCZĘSNY Ryszard, Żyrardowskie Zakłady Tkanin Technicznych,
ul. Okrzei 51, 96-300 Żyrardów, tel.sł. 20-31/5 wewn.224.
139. SZYMCZAK Elżbieta, Instytut Mechaniki Stosowanej PŁ, ul.
Stefanowskiego 1/15, 98-324 Łódź, tel.sł. 36-55-22,
wewn. 708, 231.
140. TALKOWSKA Bogumiła, Biuro Studiów i Projektów Energetycznych
"Energoprojekt", ul. Piekary 19, 60-067 Poznań, tel.sł.
222-011 wewn. 298.
141. URBANIEC Zbigniew, Śląski MOSTOSTAL-4, , ul. Wolności 191,
41-800 Zabrze, tel.sł. 71-32-21 wewn. 205.
142. URBĄSKI Jan, PP Polmozbyt, ul. Strykowska 1/5, 91-725 Łódź,
tel.sł. 78-62-14.
143. WAKSMAN Marian, Biuro Projektów Budownictwa Elektrowni i Prze-
mysłu, ul. Jasna 14/16, Warszawa, tel.sł. 27-50-81.
144. WERBIŃSKI Ryszard, Gdańskie Zakłady Nawozów Fosforowych,
ul. Kujawska 2, Gdańsk, tel.sł. 438-263, dom. 380-181.
145. WIECZOREK Karol, MPK - Zakład Ekonomiki i Informatyki, ul.
Piotrowska 147, 90-440 Łódź, tel.sł. 366355.
146. WIERZBICKI Jan, Huta Szkła Okiennego "SZCZAKOWA", ul. Kolejarny
81, 32-520 Jaworzno, tel.sł. 774-41 wewn. 120, dom. 638-00.
147. WŁODARSKI Ladysław, Biuro Projektów Przemysłu Lekkiego "BEDETE",
Pl. Zwycięstwa 2, 90-950 Łódź, tel.sł. 36-49-81.
148. WOŹNIAK Urszula, Stocznia Remontowa "Radunia", ul. Na Ostrowiu 1,
80-873 Gdańsk, tel.sł. 31-68-31 wew. 278.
149. WÓJCICKI Andrzej, Okręgowe Przedsiębiorstwo Geodezyjno-Karto-
graficzne, ul. Zwycięstwa 140, 75-613 Koszalin, tel.sł.
277-51 wewn. 170.
150. WÓJCIK Andrzej, Zakłady Metali Lekkich "Kęty", ul. Kościuszki
111, 32-650 Kęty, tel.sł. 522-51, wewn. 420.
151. WRONIECKI Zbigniew, CTH "Prema", ul. Krakowskie Przedmieście
47/51, 00-950 Warszawa, tel.sł. 26-32-01 wewn. 344.

152. ZAKYSŁOWSKI Marek, Fabryka Osprzętu Samochodowego, ul. Przybyszewskiego 99, 93-126 Łódź, tel.sł. 84-02-40 - 44. wewn. 388.
153. ZAŁĘSKA-POPOW Barbara, Zakład Zastosowań Matematyki i Informatyki Akademii Rolniczej, ul. Dr Judyma 24, Szczecin, tel.sł. 700-61 wewn. 91.
154. ZAWADZKI Wiesław, Zrzeszenie Przemysłu Ciągnikowego "Ursus", ul. Przemysłowa 14/15, 66-400 Gorzów Wlkp. tel.sł.27221,wewn.314.
155. ZIELIŃSKI Stefan, Instytut Okrętowy Politechniki Gdańskiej, ul. Majakowskiego 11/12, 80-952 Gdańsk, tel.sł. 47-21-32.
156. ZIELONKA Jerzy, Biuro Projektowo Badawcze Budownictwa Ogólnego MIASTOPROJEKT-2, ul. Wólczańska 20, Łódź, tel.sł.32-81-00, wewn. 36.
157. ZIÓŁKOWSKI Andrzej, Instytut Badań Systemowych PAN, ul. Nowelska 6, 01-447 Warszawa, tel.sł. 361901 wewn. 210.
158. ŻELIŃSKI Jerzy, Cieszyńska Fabryka Parb i Lakierów "Polifarb", 43-400 Cieszyn Markowice, tel.sł. 214-10.
159. ŻYŁŁA Romuald, Politechnika Łódzka, Inst. Inż. Chem. ul. Wólczańska 175, 90-924 Łódź, tel.sł. 36-55-22, wewn. 843.

LISTA CZŁONKÓW POROZUMIENIA UŻYTKOWNIKÓW MINIKOMPUTERA MERA-400

Lp.	Nazwa i adres instytucji	Pełnomocnik/zastępca	Tel.
1	2	3	4
1	Akademia Górniczo-Hutnicza im. Stanisława Staszica, Al. Mickiewicza 30, 30-059 Kraków	doc.dr hab. Marta Wasilewska-Radwańska dr inż. Marek Książkiewicz mgr inż. Krzysztof Palacz mgr inż. Wojciech Słota mgr inż. Janusz Miller	34-00-10 34-16-98 33-91-00 w. 39-15 33-91-00 w. 30-29 33-91-00 w. 28-55
2	Akademia Rolnicza w Szczecinie, Zakład Zastosowań Matematyki i Informatyki, ul. Dr Judyta 24, Szczecin	doc.dr hab. Zenon Woźniak mgr Barbara Załęska- mgr Antoni Miklewski	700-61 w. 91 700-61 w. 91 700-61 w. 91
3	Biuro Projektów Budownictwa Komunalnego, ul. Grunwaldzka 2, 82-300 Elbląg	mgr Piotr Czajkowski mgr inż. Roman Swięcki inż. Henryk Pudzyński	259-03 259-03 243-76
4	Biuro Projektów Budownictwa Komunalnego, ul. H. Sawickiej 27, 80-237 Gdańsk-Wrzeszcz	Roman Mysiek Ryszard Tuziemski	41-40-11 w. 13 41-40-11 w. 13
5	Bydgoskie Zakłady Przemysłu Gumowego "STOMIL", ul. Toruńska 155, 85-950 Bydgoszcz	mgr inż. Henryk Stachowski mgr inż. Jerzy Stefański mgr Andrzej Poziemski	61-16-41 w. 412 61-16-41 w. 413 61-16-41 w. 179
6	Biuro Projektów Służby Zdrowia, ul. Astrów 10, Katowice	mgr Wiesław Nowak Zofia Kremer	582-840 582-840
7	Biuro Projektów Budownictwa Ogólnego "MIASTO-PROJEKT w Kraków", ul. Kraszewskiego 36, Kraków,	Krzysztof Orłof	22-51-00 w. 385, 364
8	Biuro Projektów Budownictwa Komunalnego Plac Na Stawach 1, 30-107 Kraków,	Czarnecka Mirosława Kazińska Teresa	22-32-00 w. 264 22-32-00 w. 264

1	2	3	4
9	Biuro Projektowo-Badawcze Budownictwa Ogólnego MIASTOPROJEKT, ul. Traugutta 21/23, 90-113 Łódź,	Mieczysław Gicchocki Janusz Oleski	33-81-19 lub 32-89-20 w. 248 33-81-19 lub 32-89-20 w. 248
10	Biuro Projektowo Badawcze Budownictwa Ogólnego "Miastoprojekt-2", ul. Włocławskiego 20, Łódź	dr inż. L. Maro Włodzimierz Karozemski	3281-00 w. 36 3281-00 w. 37
11	Biuro Projektów Przemysłu Lekkiego "BEDETE" Pl. Zwycięstwa 2, 90-950 Łódź	mgr Ładysław Włodarski mgr Marek Czajkowski mgr Tomasz Stefaniak	36-48-81 36-33-44 w. 124 36-33-44 w. 113
12	Biuro Projektów Budownictwa Komunalnego ul. Tuwima 22/26, 90-002 Łódź	mgr inż. Barbara Szozepaniak mgr Marek Tynenaki	32-32-75 32-32-75
13	Biuro Studiów i Projektów Energetycznych "Energoprojekt", ul. Piekary 19, 604967 Poznań	mgr inż. Ryszard Grzybowaki inż. Edmund Łuczak	22-20-11 w. 298 22-20-11 2. 216
14	Biuro Projektów Budownictwa Wiejskiego ul. Piekary 17, 60-959 Poznań	mgr inż. Michał Gawęcki inż. Bogdan Machowiak	33-05-81 w. 570 33-05-81 w. 569
15	Biuro Projektów Budownictwa Morakiego "BIMOR" Pleco Batorego 4, 70-207 Szczecin	mgr Władysław Kowalozyk mgr inż. Mieczysław Kosecki	40-33-05 40-33-05
16	Biuro Projektów "NAFTOPROJEKT" ul. Mysiak 00-496 Warszawa,	mgr inż. Maria Krąjewska	28-40-21 w. 88
27	Biuro Projektów Budownictwa Wiejskiego ul. Juliana Nowickiego 32, 87-100 Toruń	mgr Marek Dobrowolski mgr Andrzej Kaczor	210-51 w. 58 210-51 w. 88
28	Centralne Biuro Studiów i Projektów Budownictwa Wodnego "Hydroprojekt", Oddział Gdańsk, ul. Powstańców Warszawskich 202, 80-162 Gdańsk	mgr inż. Henryk Noga mgr inż. A. Danecka-Noga	41-20-61, 41-60-80 41-20-61, 41-60-80
19	Centralny Ośrodek Badawczo-Rozwojowy Maszyn Włókienniczych POLIMATX-CENARO, ul. Wólczańska 55/59, 90-950 Łódź	mgr inż. Tadeusz Grall mgr Włodz. Jastrzębski	32-85-70 w. 243 32-85-70 w. 243

1	2	3	4
20	Centralny Ośrodek Metodyczny Poradnictwa Wychowawczo-Zawodowego MOiW, Zespół Systemów Informatycznych- ul. Nowowiejska 5, 00-643 Warszawa	mgr Barbara Aleksandrowicz Bartłomiej Biczek	29-63-59 29-63-59
21	Centralna Techniczno-Handlowa Przemysłu Precyzyjnego "Prama", ul. Krakowskie Przedmieście 47/51, Warszawa	mgr inż. J. Konczerewicz mgr A. Mierzejewski mgr Z. Wroniecki	26-32-01 w.295 26-32-01 w.295 26-32-01 w.295
22	Centralny Ośrodek Badawczo-Projektowy Budownictwa Przemysłowego "Bistyp", ul. Parkingowa 1, 00-518 Warszawa	mgr inż. Hanna Krzyszczuk mgr inż. Krzysztof Kociatkiewicz	28-94-71 28-94-71
23	Centrum Medycyny Doświadczalnej i Klinicznej Polskiej Akademii Nauk, ul. Dworkowa 3, 00-784	mgr Leszek Czerwosza mgr Joanna Kulcsa	49-74-88, 49-66-51 49-74-88, 49-66-51
24	Centralne Biuro Studiów i Projektów Budownictwa Wodnego "HYDROPROJEKT", ul. Dabois 9, 00-182 Warszawa	mgr inż. Kazimierz Nowacki	38-70-41 w.246
25.	Fabryka Osprzętu Samochodowego "POLIMOR", ul. Przybyшевskiego 99, 93-126 Łódź	mgr inż. Edward Zamysłowski	84-02-40, 84-56-84
26	Gdańskie Zakłady Nawozów Fosforowych, ul. Kujańska 2 80-958 Gdańsk	Ryszard Werbiński mgr J. Ogórkowski	438-263 438-272, 438-310
27	Główne Biuro Studiów i Projektów Przeróbki Węgla SEPARATOR, ul. Armii Czerwonej 2, 40-952 Katowice	mgr inż. Zb. Krypczyk mgr Elżbieta Piweo	586-071 w.513 586-071 w.513
28	Huta Szkła Okienno "Szczakowa", ul. Kolejarzy 31, 32-520 Jaworzno	mgr inż. Jan Wierzbicki mgr inż. Maria Perok	774-41 w.120 774-41 w.282
29	Huta Szkła Okienno "KARA", Przedsiębiorstwo Państwowe, ul. Paplińskiego 1, 97-300 Piotrków Trybunalski	mgr Wojciech Badura Jolanta Olejniczak	16-11-41 w.97 16-11-41 w.97

1	2	3	4
30	Huta Szkła Walcowanego "Jaroszwiec", ul. Kolejowa 1, 32-312 Jaroszwiec	mgr inż. Stanisław Barczyk	309-14 w.143
31	Instytut Okrętowy Politechniki Gdańskiej 80-952 Gdańsk, ul. Majakowskiego 11/12	mgr inż. Andrzej Braniecki mgr inż. Zbigniew Czerniak	47-16-43 47-17-08
32	Instytut Odlewnictwa Politechniki Śląskiej, ul. Towarowa 7, 44-100 Gliwice	mgr inż. Ewa Majchrzak mgr Włodzimierz Stefaniak	31-60-31 31-60-31
33	Instytut Informatyki Uniwersytetu Jagiellońskiego, ul. Reymonta 4/243, 30-059 Kraków	mgr Kazimierz Jojczyk mgr Mieczysław Guja	33-63-77 w.538 33-63-77 w.538
34	Instytutu Włókiennictwa, ul. Gdańska 91/93, 90-950 Łódź	mgr Jerzy Grall mgr Krystyna Białas	339-600 w.214 339-600 w.175
35	Instytut Inżynierii Chemicznej Politechniki Łódzkiej, ul. Wólczańska 175, 90-924 Łódź	mgr inż. Jan Kryczka mgr inż. Romuald Żyła	36-55-22 w.715 36-55-22 w.843
36	Instytut Maszyn Przepływowych Politechniki Łódzkiej, ul. Stefanowskiego 1/15, Łódź 90-924 Łódź	Roman Malinowski Longin Horodko	36-55-22 w.1252 36-55-22 w.1252
37	Instytut Fizyki Molekularnej PAN, ul. Smoluchowskiego 17/19, 60-179 Poznań	dr Wojciech Malinowski dr Andrzej Dozor	67-40-71 w.264 67-40-71 w.219
38	Instytut Inżynierii Lądowej Politechniki Poznańskiej, Plac Curie-Skłodowskiej 5, 60-965 Poznań	dr inż. Mieczysław Kania mgr inż. Roman Kapczyński	782-425 782-418
39	Instytut Melioracji i Użytków Zielonych, Falenty 05-550 Raszyn	mgr Henryk Michałowski mgr inż. Sławomir Maczyński	500-531 w.243 500-531 w.243
40	Instytut Chemii Fizycznej Polskiej Akademii Nauk, ul. Kasprzaka 44/52, 01-224 Warszawa	mgr inż. Grażjana Cichowska	32-32-21 w.331

1	2	3	4
41	Instytut Technologii Elektronowej Politechniki Warszawskiej, ul. Koszykowa 75, 00-662 Warszawa	mgr inż. Michał Duchnowski mgr inż. Stanisław Dmowski	21007-693 21007-906
42	Instytut Elektroenergetyki Politechniki Warszawskiej, ul. Koszykowa 75, 00-661 Warszawa	mgr inż. Andrzej Cichy	218688
43	Instytut Miskich Temperatur Badań Strukturalnych PAN, Pl. Katedralny 1, 50-950 Wrocław.	dr Ludwik Biegała inż. Kazimierz Nawrot	22-10-71 w. 72 22-10-71 w. 72
44	Komedna Wojewódzka Straży Pożarnych, ul. Bema 17, 82-300 Elbląg	mgr inż. J. Przybyłek	276-91 w. 213
45	Kopalnia Węgla Kamiennego "SIERSZA", ul. Kopalniana, 32-541 Trzebinia	mgr inż. Marek Kosowski mgr Aleksander Krzemień mgr inż. Andrzej Kozioł	Trzebinia 9 w. 542 Trzebinia 9 w. 251 Trzebinia 9 w. 251
46	Krakowskie Biuro Projektowo-Badawcze Budownictwa Ogólnego, Al. Powstańców Warszawy 10, 31-549 Kraków	mgr inż. Anna Dymek mgr Barbara Piotrowska	11-87-33 w. 347 11-87-33 w. 347
47	Międzyresortowy Instytut Techniki Radiacyjnej, Politechnika Łódzka, ul. Wróblewskiego 15, 93-590 Łódź	mgr inż. Mirosław Póret mgr inż. Dorota Świątła mgr inż. Mirosław Sopek	36-55-22 w. 12-33 36-55-22 w. 12-33 36-55-22
48	Ośrodek Analizy Wartości, Inżynierska Spółdzielnia Pracy, ul. Skupska 28, 80-302 Gdańsk	mgr inż. Andrzej Dobrzański	52-02-57
49	Okręgowy Szpital Kolejowy, Pracownia Aparatury Medycznej, ul. Medyków 5, 40-760 Katowice	inż. Marian Pustolnik Maré Krzysztof	525-011 w. 22 525-011 w. 22
50	Okręgowe Przedsiębiorstwo Geodezyjno-Kartograficzne w Koszalinie, ul. Zwycięstwa 140, 75-613 Koszalin	mgr inż. Andrzej Biełkowski mgr inż. Andrzej Wójcicki mgr inż. Maria Urbanska	277-51 w. 121 277-51 w. 170 277-51 w. 168
51	Ośrodek Badawczo-Rozwojowy Systemów Automatyki, ul. Czerwonej Armii 66/72, 61-807 Poznań	dr inż. Andrzej Wachowski mgr inż. Andrzej Korytek mgr inż. Małgorzata Piasoik	53437 53437 45681

1	2	3	4
52	Okręgowe Przedsiębiorstwo Geodezyjno Kartograficzne, ul. Zwierzyniecka 10, 60-813 Poznań	B. Biełkowska J. Łykowski	472-41 w. 27 472-41
53	Ośrodek Informatyki Urzędu Wojewódzkiego, ul. Geodetów 1, 39-400 Tarnobrzeg	Jacek Skorupski Grzegorz Pańkowski	22-35-49 22-35-49
54	Ośrodek Badawczo-Rozwojowy Elektronicznej Aparatury Medycznej, ul. Wolności 345a, 41-800 Zabrze	inż. M. Czerwiński mgr J. Kręcioch mgr J. Sledzińska	71-64-21-9 w. 255 71-64-21-9 w. 255 71-64-21-9 w. 255
55	Ośrodek Badawczo-Rozwojowy Metrologii Elektrycznej, ul. Sulechowska 1, 65-022 Zielona Góra	mgr Stanisław Dziadura mgr inż. Urszula Bereza	65036 w. 5371 65036 w. 5434
56	PKP Oddział Geodezyjny, Al. Rożdżeńskiego 1, Katowice	inż. Jerzy Gottwald mgr Wojciech Sobczyk	575597 575597
57	Politechnika Poznańska, Instytut Techniki Ciepłej i Silników Spalinowych, Zakład Techniki Ciepłej, ul. Piotrowa 3, 60-965 Poznań	dr inż. Bohdan Deptuła mgr Bogumił Domalanus mgr Wacław Gołaś	782-215 782-215, 782-201 782-201
58	Przemysłowe Centrum Optyki w Budowie, ul. Ostrobramska 75, Warszawa	Grzegorz Głowacz Jerzy Kutra	13-73-94 13-73-94
59	Politechnika Wrocławska, Instytut Chemii Organicznej i Wzyczonej I-4, Ul. Wybrzeże Wyspiańskie-go 27, 50-370 Wrocław	dr inż. Ryszard Radomski mgr inż. Mirosław Biernat dr inż. Maria Radomska	20-24-12 lub 20-3783 20-37-83 20-24-12, 20-37-83
60	Polskie Górnictwo Małtowe i Gazownictwo, Górnośląski Okręgowy Zakład Gazownictwa, ul. Gwardii Ludowej 11, 41-800 Zabrze	mgr inż. Tadeusz Kochel Teodor Gorczyk mgr inż. Joanna Garstka-Macko- wiak	71-52-21 w. 5415 71-52-21 w. 5415 71-52-21 w. 5415
61	Rybnicki Zakład Przetwarzania, ul. Wiejska 7, 44-200 Rybnik	mgr inż. Dorota Płaczek Wiesława Malawana	26451/81 26451/81
62	Śląska Akademia Medyczna, Centrum Elektronicznego Przetwarzania Danych, ul. Medyków, 40-752 Katowice - Ligota	dr inż. Janusz Pietkiewicz mgr Andrzej Horak mgr Dorota Nowak	527-081 w. 1466 527-081 w. 1466 527-081 w. 1466

1	2	3	4
63	Słaskie Przedsiębiorstwo Konstrukcji Stalowych i Urządzeń Przemysłowych MOSTOSTAL M-4, ul. Wolności 191, 41-800 Zabrze	Zbigniew Urbaniec Marek Duszański Stanisław Kędzierski	7132-21 w. 201 7132-21 w. 202 7132-21 w. 202
64	Wojskowy Instytut Higieny i Epidemiologii im. gen. Karola Kaczkowskiego, ul. Kozielecka 4, 01-163 Warszawa	mgr inż. Zbigniew Drzewiecki	250441 prosid 63-112
65	Warszawskie Przedsiębiorstwo Geodezyjne, ul. Nowy Świat 276, 00-955 Warszawa	mgr inż. Z. Leszczewicz mgr inż. J. Adamczewska mgr inż. O. Butowski	21-44-61 w. 259 21-44-61 w. 321 21-44-61 w. 259
66	Warszawska Spółdzielnia Mieszkańcowa, ul. Krasin- skiego 16, 01-581 Warszawa	Zdzisław Jagodziński mgr inż. Michał Brzeski mgr Jerzy Peszek	39-51-00 39-94-54 w. 54 39-94-54 w. 54
67	Wrocławskie Biuro Projektowo-Badawcze Budow- nictwa Przemysłowego, ul. Świdnicka 19, 50-960 Wrocław	Witold Czarnacki Grażyna Bednarek Roman Barecki	380-41 w. 252 380-41 w. 253 380-41 w. 253
68	Wojewódzkie Biuro Projektów, ul. Wolności 286, 41-801 Zabrze	mgr inż. Karol Bresler inż. Andrzej Pisiewicz	71-20-21 w. 10 71-20-21 w. 84
69	Wojewódzki Ośrodek Informatyki w Bkiermieniowcach Z/S w Żyrardowie, ul. Jaktorowska 20/22, 96-300 Żyrardów	Renata Opłatek Andrzej Malinowski	31-12 31-12
70	Zakłady Automatyki Chemicznej "METALCHEM" ul. Chorzowska 44, 44-101 Gliwice	inż. H. Kwapiński Z. Nowak mgr inż. B. Pizoń	31-64-41/102 31-64-41/102 31-64-41/102
71	Zakłady Metali Lekkich "Kęty", ul. Kościuszkii 111, 32-650 Kęty	mgr inż. Eugeniusz Borowski inż. Danuta Ciałowicz mgr Wiktor Kralka	522-51, w. 217 522-51 w. 464 522-51 w. 154

72	Biurow Projektów Budownictwa Elektrowni i Przemysłu "Elpro", ul. Jasna 14/16, 00-950 Warszawa	Marian Wakoman	27-50-81
73	Centrum Badańczo-Wdrożeniowe "MERCOPI"- Mera-Przełaj, ul. Poczty 19, 04-994 Warszawa	Szancer Wojciech mgr inż.	25-56-40
74	Muta Szkoła Okienego "Sandomierz" ul. Portowa 24, 27-600 Sandomierz	Kocój Jan Kucharski Józef	30-41 w.143 30-41 w.143
75	Przedsiębiorstwo Zagraniczne "Amepol" ul. Plac Żelaznej Bramy 1, 00-136 Warszawa	Jerzy Dżoga Jadwiga Stecura	20-34-75
76	Stocznia Remontowa "Radunia" ul. Na Ostrowiu 1, 80-873 Gdańsk	Woźniak Urszula Bobcow Andrzej	31-69-31 w.278 31-68-31 w.278
77	Zakład Informatyki Spółdzielni Mieszkaniowych Spółdzielnia Osób Prawnych, ul. Kliny 2, 31-465, Kraków	Zbigniew Bauer Zygmunt Rewilak Jan Dudek	11-94-67 11-94-67 11-94-67
78	Zakład Ekonomiki i Informatyki Miejskiego Przedsiębiorstwa Komunikacyjnego, ul. Piotrkowska 147/149, Łódź	mgr Krystyna Pietrzak mgr Lucjan Durkiewicz mgr Konstanty Owczarek	3648-12, 36-15-60 32-19-32 3663-55
79	Zespół Opieki Zdrowotnej, ul. Kamieniec 10, 34-500 Zakopane	mgr inż. Anna Smol inż. Kazimierz Slusarczyk	20-21 w. 288 20-21 w. 288
80	Żyrardowskie Zakłady Tkanin Technicznych, ul. Okrzei 51, 96-300 Żyrardów	Ryszard Szczepny Adam Rohlik Jacek Owczarek	20-31 do 35 w.284 20-31 do 35 w.284 20-31 do 35 w.294

