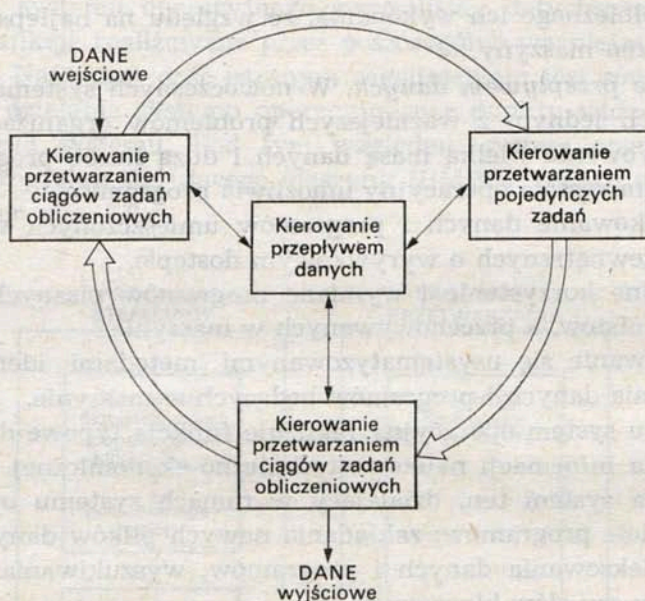


z biblioteki programów do wykonania podobnych, ale jednak różnym celom służących zadań.

Wzajemne zależności funkcji systemu operacyjnego w zakresie kierowania przetwarzaniem ciągów zadań obliczeniowych i pojedynczych zadań obliczeniowych oraz przepływem danych — ilustruje ryc. 3.10.



Ryc. 3.10. Schemat wzajemnych zależności trzech funkcji kierowniczych systemu operacyjnego maszyny typu IBM 360

Dla maszyn serii ICL 1900 system operacyjny znany jest pod nazwą GEORGE. Dla modeli 1904, 1905, 1909 wykorzystywany jest GEORGE 1, dla modeli 1909, 1905 (większych zestawów) i 1906, 1907 (mniejszych zestawów) stosuje się GEORGE 2. Natomiast GEORGE 3 i 4 przeznaczone są do dużych instalacji ICL 1906 i 1907.

System operacyjny GEORGE nie dysponuje możliwością kierowania zadaniami, lecz tylko umożliwia kierowanie przebiegiem operacji ogniwa przetwarzania (job).

3.8. Niektóre programy uniwersalne

Programy uniwersalno-usługowe (standardowe) pod względem sposobu ich wykorzystania dzielą się na:

a) podprogramy włączone do programów użytkowych, specjalnie zredagowanych,

b) samodzielne programy zapewniające całkowitą realizację określonych typowych przebiegów,

c) samodzielne programy pomocnicze.

Podprogramy są sekwencją rozkazów, które rozwiązują funkcje autonomicznie występujące w całym programie, a których częstotliwość występowania uzasadnia zredagowanie uniwersalnego podprogramu. Przy każdorazowej konieczności rozwiązania określonej funkcji następuje skok z programu głównego do podprogramu, a po jego wykonaniu ponowny powrót do programu głównego.

Do typowych podprogramów można zaliczyć:

- tłumaczenie kodów stosowanych w zewnętrznych nośnikach informacji na kod stosowany wewnątrz maszyny.

- tłumaczenie kodu maszyny na kod stosowany w zewnętrznych nośnikach informacji (perforacja i druk),

- realizację operacji mnożenia i dzielenia, jeżeli lista rozkazów maszyny nie obejmuje tych funkcji,

- normalizowanie liczb przy konwersji systemu stałoprzecinkowego na zmiennoprzecinkowy i odwrotnie,

- rozwiązywanie funkcji trygonometrycznych, logarytmów, pierwiastków, macierzy itp.,

- zmiany systemów monetarnych (np. przeliczenia szterlingowe) lub systemów miar (np. zamiana systemu calowego na metryczny).

Programy typowych przebiegów obejmują klasyczne przebiegi występujące na ogół we wszystkich zastosowaniach; dotyczą one:

- przenoszenia informacji zapisanych na nośnikach w relacjach: karta — taśma lub karta — dyski magnetyczne (w obu kierunkach); taśma dziurkowana (karta) — dyski magnetyczne (w obu kierunkach); taśma (karta, dyski magnetyczne) — tabulogramy,

- sortowania,

- dobierania,

- wydruku,

- modyfikacji zapisów na taśmie (kartach, dyskach magnetycznych).

Programy pomocnicze obejmują takie funkcje standardowe, które spełniają uzupełniającą, pomocniczą rolę w stosunku do głównych, zasadniczych przebiegów oraz operacji przetwarzania; dotyczą one:

1. Fazy uruchamiania programów w zakresie:

- tłumaczenia programów zredagowanych w autokodach na język wewnętrzny maszyny (tzw. translatory),

- zapisywania w pamięci zewnętrznej danych próbnych,

- zapisywania w pamięci zewnętrznej programów,

- wydruku zawartości pamięci operacyjnej,

- wydruku zawartości pamięci zewnętrznej,

- wydruku wybranych programów zapisanych w pamięci zewnętrznej,

- wydruku realizacji programu rozkaz po rozkazie wraz z zawartością wybranych rejestrów (tzw. *pas à pas*),
- programowanego zatrzymywania programu po wykonaniu określonego rozkazu, który mógł być np. iterowany określoną ilość razy,
- wydruku tzw. spisu treści określonej części pamięci zewnętrznej, np. nazwy kartoteki i liczby bloków, symboli sterując na taśmie; wydruku np. pierwszych 100 znaków każdego bloku,
- automatyzacji testowania programów na zbiorach próbnych danych.

2. Fazy eksploatacji programów w zakresie:

- wprowadzania danych początkowych, które są niezbędne w dalszej fazie eksploatacji, np. tablic działań algebraicznych (do maszyn znakowych bezakumulatorowych), tablic kodu drukarki, podprogramów tłumaczenia kodów, wprowadzania programów,
- kontroli kolejności i warunków wykonywania programów w ramach ustalonego codziennego harmonogramu przetwarzania, w zależności od warunków kalendarzowych i innych, pobierania z biblioteki umieszczonej w pamięci zewnętrznej poszczególnych programów przewidzianych do wykonania.

Znajomość biblioteki programów uniwersalnych jest warunkiem prawidłowego projektowania technologii maszynowego przetwarzania.

W dalszej kolejności omówimy bliżej programy dotyczące modyfikacji zbiorów informacji na taśmach oraz automatycznego testowania.

3.8.1. Program modyfikacji zbiorów informacji zapisanych w pamięci zewnętrznej

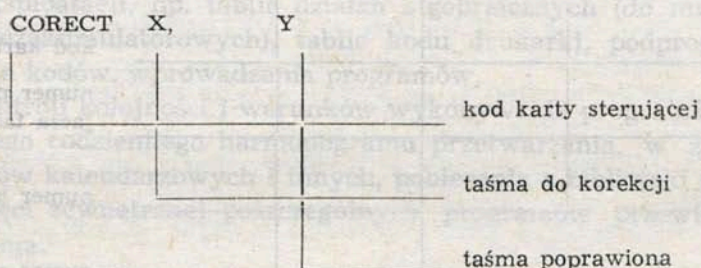
Program ten wykonuje funkcje potrzebne do grupowania lub przegrupowywania informacji zapisanych w pamięci zewnętrznej maszyny, np. na taśmach, kartach, dyskach magnetycznych. Maszyna NCR 315 posiada taki program pod nazwą LIBRARIAN (bibliotekarz); w maszynach ICT 1300 i innych oznacza się najczęściej ten program nazwą FILE MAINTENANCE, czyli prowadzenie kartotek.

Do głównych funkcji takiego programu należy (na przykładzie taśm magnetycznych):

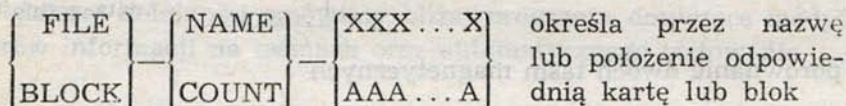
- kontrola wpisywania metryk na taśmę magnetyczną,
- porównywanie jednej taśmy magnetycznej z drugą,
- kopiowanie taśm magnetycznych,
- korekta zawartości taśmy magnetycznej według informacji zawartych w innym zbiorze,
- zmiana pozycji na taśmie magnetycznej,
- ustawienie taśmy magnetycznej bezpośrednio przed lub za kartoteką albo żądanym blokiem (np. w wypadku zapisania na jednej taśmie magnetycznej kilku różnych zbiorów informacji-kartotek),
- przewijanie taśm magnetycznych,

Y — nr jednostki taśmy magnetycznej, na którą następuje przekopiowanie,
 FILE-NAME — nazwę zbioru na taśmie magnetycznej, od którego następuje kopiowanie,
 BLOCK-COUNT — kopiowanie na zasadzie liczenia bloków i znalezienia takiego, od którego następuje kopiowanie.

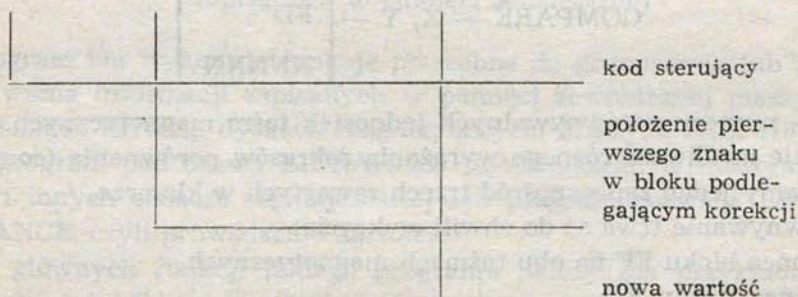
- d) korygowanie zapisu, które odbywa się przez określenie:
- numeru kolejnego bloku i miejsca w danym bloku,
 - wartości danego argumentu jako adresu, dzięki któremu odszukuje się pozycję



Wybieramy jedną z dwóch możliwości zawartych w klamrze:



AT — M M M M M — CORECT — XXX...X



Istnieje możliwość różnych wariantów deklaracji, polegająca na podawaniu numerów pozycji lub ich nazw.

3.8.2. Automatyzacja testowania programu

Testowanie jest na ogół najtrudniejszą fazą procesu projektowania. Na przykład pewien złożony program dla brytyjskiego systemu ubezpieczeń został napisany w IBM Data Center (Londyn) w ciągu 2 miesięcy, podczas gdy testowanie zajęło aż 7 miesięcy. Pewne ułatwienia w tym zakresie zilustrujemy na przykładzie systemu CONSOLIDATA.

Zespół programów, które umożliwiają programiście w sposób łatwy uruchamianie programów na danych próbnych, stosowany jest w dwóch fazach:

W fazie I grupuje się na tej samej taśmie magnetycznej kilka programów odpowiadających każdemu zestawowi danych próbnych oraz wszelkich niezbędnych programów standardowych.

W fazie II umieszcza się dane próbne na odpowiednich jednostkach taśm magnetycznych oraz rozpoczyna się testowanie. Przy każdorazowym zatrzymaniu na skutek błędów w programie, zostaje automatycznie wydrukowana zawartość odpowiednich pamięci oraz zestawienia wszystkich użytych taśm magnetycznych i ich pozycji (aż do momentu zatrzymania się programu). Na specjalne żądanie możliwe jest otrzymanie wyników pośrednich po wykonaniu ważniejszej partii programu lub wydruk wartości odpowiednich rejestrów i pamięci po wykonaniu każdego rozkazu. Na podstawie tak przygotowanego przez maszynę materiału programista wprowadza poprawki i ponownie przekazuje program do sprawdzenia na maszynie.

3.9. Niektóre systemy programów uniwersalnych

Podana w pkt. 3.1. ogólna struktura oprogramowania ulega pewnym modyfikacjom w miarę rozwoju metod programowania. Zmiany te wynikają z coraz szerszego stosowania uniwersalnych programów, a ostatnio całych systemów programów. Rozwój programów uniwersalnych dotyczy przede wszystkim programów usługowych i użytkowych. Systemy programów uniwersalnych można podzielić na:

- a) pakiety programów usługowych (*utility packages*),
- b) pakiety programów systemów przetwarzania danych (*application dependent packages*),
- c) pakiety programów specjalnych — niezależnych (*application independent packages*).

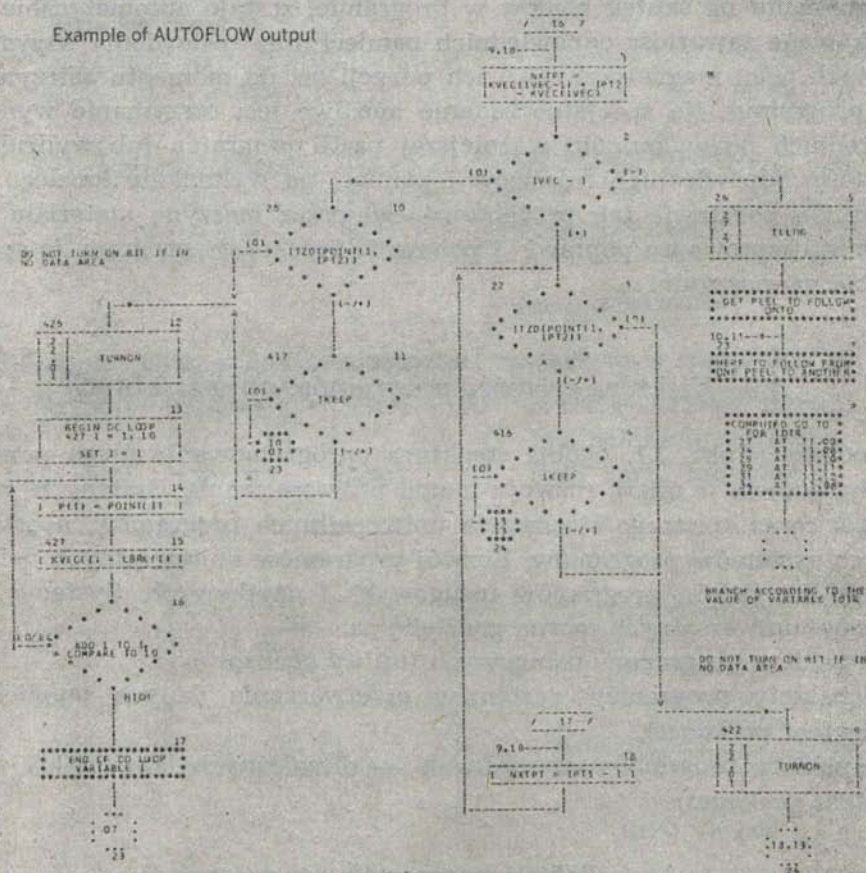
3.9.1. Pakiety programów usługowych

Oprócz wymienionych w pkt. 3.7. programów usługowych występują ponadto systemy programów, które ułatwiają opracowywanie dokumentacji technicznej systemów APD. Wyróżniają się tu dwa następujące systemy:

- a) różnego typu programy AUTOFLOW drukujące schematy blokowe na podstawie listy rozkazów w językach COBOL, FORTRAN, AUTOCODER i in. Na ryc. 3.11 podajemy przykład schematu blokowego wydrukowanego na podstawie listy rozkazów w języku FORTRAN,
- b) programy opracowujące dokumentację oprogramowania systemu.

CHART TITLE - SUBROUTINE FOLLOW

Example of AUTOFLOW output



Example of source program input

```

14  KVECT(IYC) = IPT2
15  KVECT(IYC) = IPT2
16  KVECT(IYC) = IPT2
17  KVECT(IYC) = IPT2
18  KVECT(IYC) = IPT2
19  KVECT(IYC) = IPT2
20  KVECT(IYC) = IPT2
21  KVECT(IYC) = IPT2
22  KVECT(IYC) = IPT2
23  KVECT(IYC) = IPT2
24  KVECT(IYC) = IPT2
25  KVECT(IYC) = IPT2
26  KVECT(IYC) = IPT2
27  KVECT(IYC) = IPT2
28  KVECT(IYC) = IPT2
29  KVECT(IYC) = IPT2
30  KVECT(IYC) = IPT2
31  KVECT(IYC) = IPT2
32  KVECT(IYC) = IPT2
33  KVECT(IYC) = IPT2
34  KVECT(IYC) = IPT2
35  KVECT(IYC) = IPT2
36  KVECT(IYC) = IPT2
37  KVECT(IYC) = IPT2
38  KVECT(IYC) = IPT2
39  KVECT(IYC) = IPT2
40  KVECT(IYC) = IPT2
41  KVECT(IYC) = IPT2
42  KVECT(IYC) = IPT2
43  KVECT(IYC) = IPT2
44  KVECT(IYC) = IPT2
45  KVECT(IYC) = IPT2
46  KVECT(IYC) = IPT2
47  KVECT(IYC) = IPT2
48  KVECT(IYC) = IPT2
49  KVECT(IYC) = IPT2
50  KVECT(IYC) = IPT2
51  KVECT(IYC) = IPT2
52  KVECT(IYC) = IPT2
53  KVECT(IYC) = IPT2
54  KVECT(IYC) = IPT2
55  KVECT(IYC) = IPT2
56  KVECT(IYC) = IPT2
57  KVECT(IYC) = IPT2
58  KVECT(IYC) = IPT2
59  KVECT(IYC) = IPT2
60  KVECT(IYC) = IPT2
61  KVECT(IYC) = IPT2
62  KVECT(IYC) = IPT2
63  KVECT(IYC) = IPT2
64  KVECT(IYC) = IPT2
65  KVECT(IYC) = IPT2
66  KVECT(IYC) = IPT2
67  KVECT(IYC) = IPT2
68  KVECT(IYC) = IPT2
69  KVECT(IYC) = IPT2
70  KVECT(IYC) = IPT2
71  KVECT(IYC) = IPT2
72  KVECT(IYC) = IPT2
73  KVECT(IYC) = IPT2
74  KVECT(IYC) = IPT2
75  KVECT(IYC) = IPT2
76  KVECT(IYC) = IPT2
77  KVECT(IYC) = IPT2
78  KVECT(IYC) = IPT2
79  KVECT(IYC) = IPT2
80  KVECT(IYC) = IPT2
81  KVECT(IYC) = IPT2
82  KVECT(IYC) = IPT2
83  KVECT(IYC) = IPT2
84  KVECT(IYC) = IPT2
85  KVECT(IYC) = IPT2
86  KVECT(IYC) = IPT2
87  KVECT(IYC) = IPT2
88  KVECT(IYC) = IPT2
89  KVECT(IYC) = IPT2
90  KVECT(IYC) = IPT2
91  KVECT(IYC) = IPT2
92  KVECT(IYC) = IPT2
93  KVECT(IYC) = IPT2
94  KVECT(IYC) = IPT2
95  KVECT(IYC) = IPT2
96  KVECT(IYC) = IPT2
97  KVECT(IYC) = IPT2
98  KVECT(IYC) = IPT2
99  KVECT(IYC) = IPT2
100 KVECT(IYC) = IPT2

```

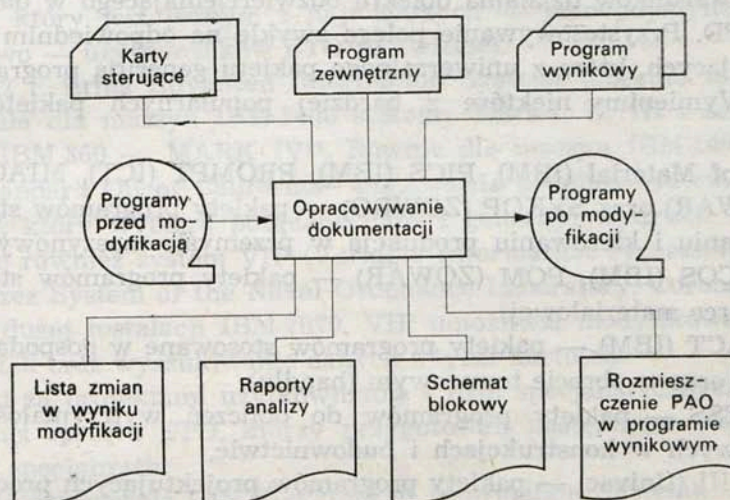
Ryc. 3.11. Przykład wydruku schematu blokowego na podstawie listu rozkazów programu ułożonego w języku FORTRAN

Tego typu programy są m.in. opracowane przez IBM o nazwie Documentation Aids Systems (1401-SE-12X), których ideę przedstawiamy na ryc. 3.12.

Zastosowanie programu opracowującego dokumentację umożliwia:

1. Modyfikowanie programów systemowych, w wyniku którego powstaje:

- zmodyfikowana taśma magnetyczna z programami zewnętrznymi,
- nowa talia kart z programami zewnętrznymi,



Ryc. 3.12. System programów opracowujących dokumentację programów (IBM-Documetation Aids System)

— lista zmian wprowadzanych w toku modyfikowania:

- lista zmodyfikowanych programów zewnętrznych,
- lista wyłączonych sekwencji rozkazów,
- lista włączonych sekwencji rozkazów,
- lista informacji dla operatora.

2. Analizowanie programów, które prowadzi do wydruku następujących informacji:

- wykazu krzyżowego⁸ etykiet,
- wykazu oznaczeń diagnostycznych (przy tłumaczeniu)⁹,
- wykazu częstotliwości operacji,
- listy informacji dla operatora.

3. Wydruk schematów blokowych w następujących przekrojach:

⁸ Polegający np. na podaniu na specjalnym tabulogramie miejsc w programie, gdzie występują te same etykiety.

⁹ W fazie tłumaczenia programu można otrzymać specjalny tabulogram z listą rozkazów programu zewnętrznego i z zaznaczonymi miejscami błędnie użytych rozkazów lub poszczególnych elementów rozkazu.

- schemat blokowy programu wynikowego,
- schemat blokowy programu zewnętrznego,
- wykaz etykiet ważniejszych partii programu wraz z podaniem numeracji stron, na których znajdują się szczegółowe schematy.

3.9.2. *Pakiety programów systemów przetwarzania danych*

Dla pewnych najbardziej typowych zastosowań ETO opracowano pakiety programów uniwersalnych, które można przystosowywać do konkretnych warunków działania obiektu odzwierciedlającego w danym systemie APD. Przystosowywanie polega zwykle na odpowiednim doborze kart sterujących, które z uniwersalnego pakietu generują programy wynikowe. Wymienimy niektóre z bardziej popularnych pakietów tego typu:

a) Bill of Material (IBM), PICS (IBM), PROMPT (ICT), MIACS (GE), POP (ZOWAR) oraz SYKOP (ZOWRO) — pakiety programów stosowane w planowaniu i kierowaniu produkcją w przemyśle maszynowym,

b) MINCOS (IBM), POM (ZOWAR) — pakiety programów stosowane w gospodarce materiałowej,

c) IMPACT (IBM) — pakiety programów stosowane w gospodarce materiałowej oraz w obrocie towarowym (handlu),

d) STRESS — pakiety programów do obliczeń wytrzymałościowych i projektowych w konstrukcjach i budownictwie,

e) APT III (Univac) — pakiety programów projektujących proces technologiczny obróbki skrawaniem na obrabiarkach sterowanych programowo,

f) GOCAP (Univac) — pakiety programów wykorzystywane przy analizie obwodów elektronicznych,

g) OMEGA/LP (Univac) — pakiety programów do obliczeń z zakresu programowania liniowego,

h) PERT, CPM, RAMPS i in. — pakiety programów do analizowania sieci czynności i zdarzeń występujących w procesach projektowania większych przedsięwzięć inwestycyjnych,

i) KWIC, KWOC, GIPSSY, MEDLARS, SMART, MUSSIL, IGA (opracowany przez CIINTE i ZOWAR), ARDIS, APIS i in. — pakiety programów stosowane w procesach wyszukiwania informacji, w szczególności informacji naukowo-technicznej i ekonomicznej i inne.

3.9.3. *Pakiety programów specjalnych*

Projektowanie pakietów programów specjalnych przeżywa obecnie okres wzmożonego rozwoju. Na razie opracowywane są przede wszystkim dla zagadnień tworzenia i reorganizowania kartotek (*management file*). Tendencja ta wynika z szerokiego stosowania pamięci masowych o wy-

rynkowym dostępie. Programowanie przetwarzania z wykorzystaniem tych pamięci jest stosunkowo złożone i pracochłonne. Z tego względu konieczne stało się usprawnienie programowania.

Pierwszy tego typu pakiet został opracowany w 1957 r. przez GE-Hanford, Washington, Atomic Products Plan¹⁰ dla maszyny IBM 702. Pakiet ten usprawniał zakładanie kartotek. W 1961 r. w Advanced Information Systems — Douglas Aircraft opracowano GIRLS (*Generalized Information Retrieval and Listing System*) dla maszyny IBM 7090 (zainstalowanej w późniejszym oddziale General Electric — Missile and Space System Division, który jest jednym z największych — jako pojedyncze przedsiębiorstwo — użytkowników ETO na świecie).

W 1962 r. firma Advanced Information Systems (obecnie Informatics) opracowała dla maszyn IBM 1400 systemy Mark I, II, III i ostatnio dla maszyn IBM 360 — MARK IV¹¹. Równie dla maszyn IBM 1401/1460 — firma Standard Oil of California¹² opracowała podobny system o nazwie SELECT, który wybiera pożądane dane i generuje wydruki. Na uwagę zasługuje również system VIP (*Variable Information Processing*) opracowany przez System of the Naval Ordnance Laboratory, Corona, California, dla dużej instalacji IBM 7070. VIP umożliwia modyfikowanie aż do 40 kartotek oraz wyszukiwanie danych z tyluż kartotek.

W ślad za projektami użytkowników i firm specjalistycznych podążyli producenci sprzętu ETO, którzy przygotowali następujące pakiety programów specjalnych:

— IBM opracowała GIS (*Generalized Information System*), który skierowano do użytku w 1965 r. w maszynach IBM 360. W 1968 r. opracowano kolejne wersje do przetwarzania wielozadaniowego. GIS wymaga 128 K pamięci operacyjnej. Wypełnianie kart sterujących jest możliwe w języku symbolicznym.

— SDS opracowała w 1965 r. system o nazwie MANAGE z przeznaczeniem dla maszyn SDS 9300 i serii 900. Wymagana pojemność pamięci operacyjnej — 16 K. Jest to bardzo elastyczny system dla maszyn małych, który równocześnie może przetwarzać 20 poleceń użytkownika formułowanych w języku „SDS business language”.

— Control Data również w 1965 s. udostępniła INFOL (*Information Oriented Language*) dla maszyn CDC 3600, 3800, który wymaga pamięci operacyjnej o pojemności 64 K. Polecenia użytkownika należy formułować w języku symbolicznym COMPASS.

— Univac udostępnił w 1966 r. IMRADS (*Information Management Retrieval and Dissemination System*) do maszyn Univac 1108 o pojem-

¹⁰ Por. Mc Gee, W. C., *Generalization-Key to Successful Electronic Data Processing*, „Journal of ACM” 1959, January.

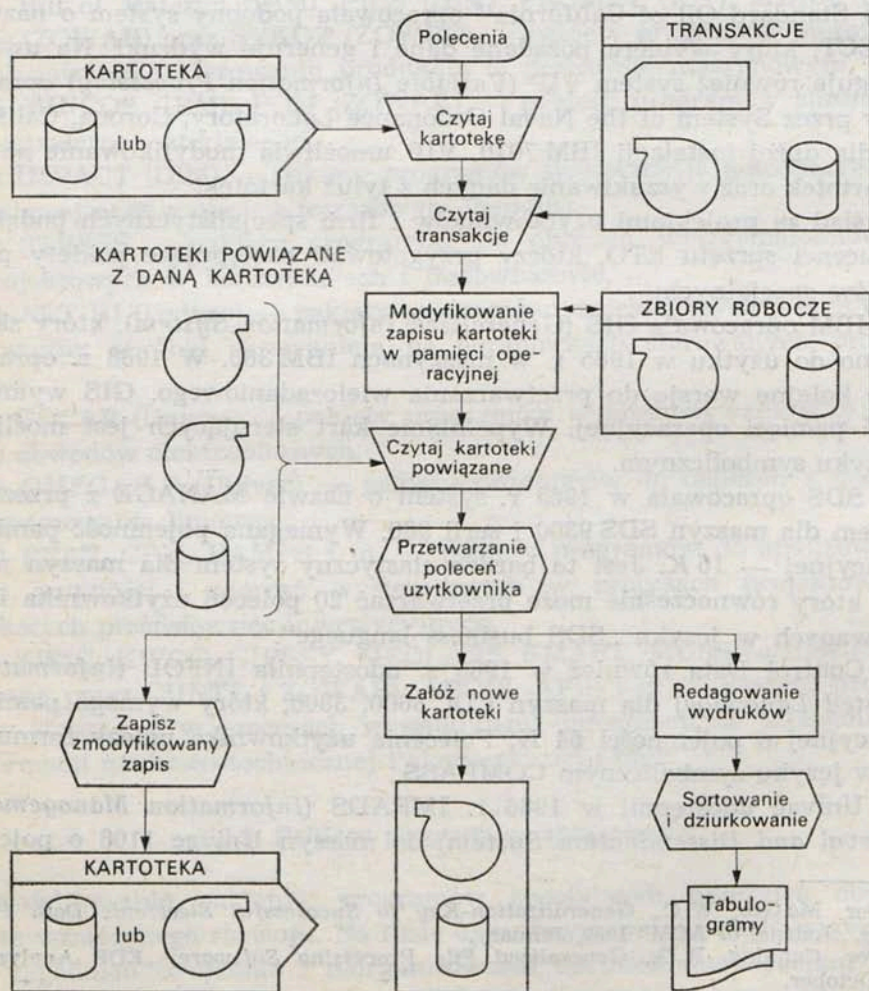
¹¹ Por. Canning, R. G., *Generalized File Processing Software*, „EDP Analyzer” 1965, October.

¹² Por. R. V. Head, *Management Information Systems, A critical Appraisal*, „Data-mation” May 1967.

ności pamięci operacyjnej 128 K. Polecenia użytkownika formułowane są w języku COBOL.

— NCR ma opracowany system BEST (*Business Electronic Data Processing Systems Technique*), który od 1964 r. stosowany jest w maszynach NCR 315. Geneza jego wywodzi się z 1962 r., w którym pierwsza wersja została opracowana dla maszyn NCR 304. BEST realizuje 55 funkcji definiowanych w języku NEAT. Wymagana pojemność pamięci operacyjnej wynosi 10 K.

— SDC udostępniła w 1966 r. system TDMS (*Time Shared Data Management System*) dla maszyny IBM 360/50, który zaliczany jest do najbardziej uniwersalnych i wygodnych w użytkowaniu. Umożliwia on modyfikowanie bardzo dużych zbiorów, zorganizowanych w banku wspólnych danych podstawowych. Polecenia formułowane są w języku użytkownika (JOVIAC).



Ryc. 3.13. MARK IV — System reorganizacji kartotek

— Firma Application Software udostępniła w 1967 r. system ASI-ST do maszyn IBM 360, SDS Sigma 5/7, RCA Spectra 70. Wymagana pojemność pamięci operacyjnej w zależności od rodzaju maszyny wynosi 16—64 K. Polecenia użytkownika formułowane są w specjalnie zaprojektowanym, łatwym języku.

Na ryc. 3.13 przedstawiamy działanie systemu MARK IV, który spełnia następujące funkcje:

— wczytywania danych z kart i taśm magnetycznych oraz z pamięci o wyrwykowym dostępie,

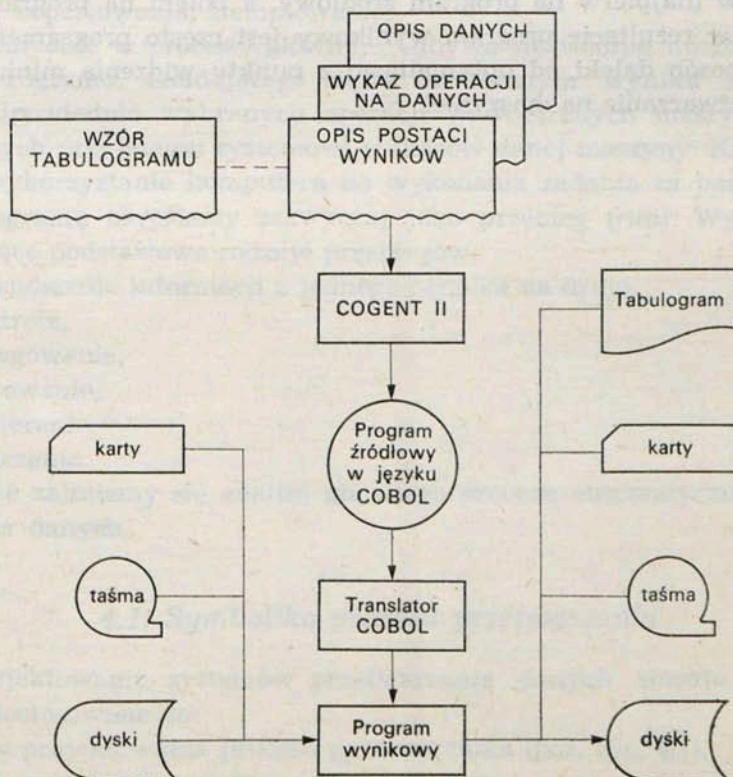
— zakładania kartotek na kartach, taśmach magnetycznych i w pamięciach o wyrwykowym dostępie; zapisy mogą być stałe lub zmienne, dostęp może być sekwencyjny lub sterowano-wyrwykowy,

— modyfikowania kartotek (wycofywanie pamięci lub ich uzupełnianie),

— aktualizowania kartotek (maksimum do czterech powiązanych ze sobą),

— wyszukiwania zapisów i wybierania danych z zapisów,

— obliczania wyników i ich drukowania lub przenoszenia do pamięci masowej,



Ryc. 3.14. GOGENT II — System reorganizacji zbiorów

- przygotowywania wydruków przez sortowanie i przegrupowywanie,
- sumowania wielopoziomowego wyników,
- drukowania wyników w postaci zawierającej dobrze zredagowane wiersze, kolumny, nagłówki, numerację stron itp.,
- tworzenia nowych kartotek z kartotek już istniejących itp.

Na ryc. 3.14 przedstawiamy system reorganizacji zbiorów — COGENT II opracowany w 1968 r. przez Computer Sciences Corporation. System ten spełnia podobne funkcje jak system Mark IV. Wymagania odnośnie do funkcji przetwarzania wypełniane są w bardzo prosty sposób na czterech formularzach: wzór tabulogramu, opis danych, wykaz operacji na danych i opis postaci wyników. Następnie pakiet programów COGENT II generuje program źródłowy w języku COBOL. Program ten może być przetwarzany na każdej maszynie matematycznej; o ile ma ona translator COBOL, który generuje program wynikowy.

Należy podkreślić, że pakiety programów specjalnych znacznie skracają okres projektowania SPD; są wygodne w stosowaniu, zwykle uniezależniają projektanta czy użytkownika od konkretnego modelu maszyny matematycznej, o ile ten np. ma translator do programu źródłowego wygenerowanego przez dany pakiet programów. Ze względu na daleko posuniętą uniwersalność pakietów programów i dwuszczeblowe tłumaczenie programów (najpierw na program źródłowy, a potem na program wynikowy) — w rezultacie program wynikowy jest często programem ułożonym w sposób daleki od optymalnego z punktu widzenia minimalizacji czasu przetwarzania na maszynie.