

Reasumując porównanie można stwierdzić, że programowanie w języku COBOL jest krótsze prawie 3-krotnie, uruchamianie zaś jest również krótsze i to 4-krotnie. Obciążenie EMC w obu wypadkach jest porównywalne. Natomiast wypełnienie PAO w języku COBOL jest o 30% większe, jak również czas przetwarzania może być o 10% dłuższy.

Przy wyborze języka programowania istotna jest wymiennność programów między różnymi maszynami, w wypadku awarii.

Istnieje przekonanie, że programy w zakresie przetwarzania danych mogą być układane wyłącznie w językach wyspecjalizowanych. Okazuje się jednak, że wiele programów z tej dziedziny zastosowań można redagować w językach przystosowanych do obliczeń numerycznych, o czym świadczy stosunkowo szerokie stosowanie języka FORTRAN IV.

W Zakładzie Elektronicznej Techniki Obliczeniowej (ZOWAR Warszawa) wykonano wiele programów na maszynie typu ICT 1300 (przy współpracy z taśmami magnetycznymi) w języku MAC, wybitnie przystosowanym do obliczeń numerycznych. W wypadkach gdy problem przetwarzania danych alfanumerycznych nie jest podstawowy, może być polecane wykorzystanie prostych autokodów.

### 3.6. Programy tłumaczące

Translatory są bardzo złożonymi i długimi programami, np. translator COBOL liczy średnio kilkadziesiąt tysięcy rozkazów w zależności od rodzaju maszyny. Przy ocenie translatora bierze się pod uwagę minimalną konfigurację zestawu komputera (na którym można realizować przetwarzanie w danym języku programowania), czas tłumaczenia oraz rozwiązywania, a także system diagnostyczny oceniający prawidłowość ułożenia programu.

Przedstawimy ogólną charakterystykę translatorów COBOL i ich głównych producentów rynku europejskiego.

ICL dysponuje dwoma kompilatorami: COMPACT COBOL i COBOL. COMPACT COBOL (wersja uproszczona) został uruchomiony dla maszyn ICL 1900 z czterema jednostkami taśm magnetycznych i z pamięcią operacyjną o pojemności 8192 słów. Uproszczona wersja COMPACT zawiera minimum listy zatwierdzonej przez Departament Obrony Stanów Zjednoczonych (mniej więcej połowę) oraz rozkaz ENTER (wprowadź do programu w COBOL rozkazy, np. w autokoderze), który ułatwia pisanie wstawek w niższych językach i rozkaz USAGE (zastosuj), który umożliwia przetwarzanie danych szterlingowych oraz w zapisie dwójkowym. Pełna wersja języka COBOL dostępna będzie dla 16 384 słów 24-bitowych pamięci operacyjnej i czterech jednostek taśm magnetycznych.

Honeywell dysponuje czterema wersjami COBOL — B, D, H, I. Wersja B wymaga 8192 znaków pamięci operacyjnej, trzech jednostek taśm



magnetycznych oraz wolnego urządzenia wejściowo-wyjściowego. Wersja ta zbliżona jest do COMPACT, lecz nie zawiera rozkazów ENTER i USAGE. Wersja D obejmuje prawie całą listę DOD, wymaga 16 384 komórek pamięci operacyjnej oraz czterech jednostek taśm magnetycznych. Wersja ta nie zawierała rozkazu: SORT (sortuj), RETURN (pozwóć), RELEASE (uwolnij), REPORT WRITER (pisz zestawienie), DEFINE, COMPUTE, READ INTO (wczytaj do), WRITE FROM (wypisz z), WRITE AFTER ADVANCING (pisz po wysuwie papieru na drukarce), RENAMES (zmień nazwę), DUMP (przechowaj), GO TO DEPENDING (skocz do zależności od ...). Wadą tej wersji jest możliwość adresowania tylko do 32 000 znaków pamięci operacyjnej. Dla większych pamięci stosowana jest wersja H, która wymaga 32 768 znaków pamięci oraz czterech jednostek pamięci taśmowej. Wersja H dysponuje tymi rozkazami, które nie zostały włączone do wersji D. Honeywell nie dysponuje na razie systemem COBOL dla dysków.

IBM dysponuje translatorami dla maszyn zarówno z pamięciami taśmowymi, jak i pamięciami dyskowymi. Dostępne są dwie wersje translatorów: E i F. Wersja E wymaga 131 072 bajtów pamięci operacyjnej oraz dwóch jednostek dyskowych. Dla IBM 360/30 w wersji taśmowej minimalna pojemność pamięci operacyjnej wynosi 16 384 bajty. Jednakże tak mała pamięć nie zapewnia efektywnej translacji; praktycznie wymagana byłaby pojemność pamięci równa 32 768 bajtom. Dla wersji dyskowej wymagana jest pojemność pamięci 32 768 bajtów oraz dwie jednostki

	IBM COBOL-F
	479.419
	IBM COBOL-E (dyski)
	286.147
Honeywell COBOL-I	
275.412	
	IBM COBOL-E (taśmy)
	254.400
Honeywell COBOL-H	
181.800	
Honeywell COBOL-D	
150.720	
	ICT COMPACT COBOL
	146.400
Honeywell COBOL-B	
122.244	

Ryc. 3.3. Ceny (w dol. USA) najtańszych zestawów komputerów różnych firm, umożliwiających stosowanie różnych wersji języka COBOL



dyskowe model 2311. Trzecia wersja przeznaczona jest do konfiguracji maszyny z pamięcią taśmowo-dyskową, gdzie dla dyskowego systemu operacyjnego (DOS) pojemność pamięci operacyjnej powinna wynieść 32 768 bajtów. Dla pełnego systemu operacyjnego pojemność pamięci operacyjnej powinna być dwukrotnie większa. Wersja E IBM odpowiada wersji H Honeywell. Wersja F przeznaczona jest dla dużych zestawów maszyn.

Wykaz diagnostyczny programu ułożonego według wersji COBOL IBM drukowany jest po wylistowaniu programu. Natomiast ten sam wykaz dla programów COBOL opracowanych przez ICT i Honeywell może być drukowany stopniowo, w miarę tłumaczenia programu źródłowego.

Ceny najtańszych zestawów maszyn różnych firm, umożliwiających stosowanie różnych wersji COBOL przedstawiamy na ryc. 3.3<sup>6</sup>. Czas tłumaczenia mierzony jest liczbą instrukcji wynikowych przetłumaczonych w ciągu minuty. Dla różnych modeli maszyn firm Honeywell i ICT czasy tłumaczenia różnych wersji dialektów COBOL przedstawiamy w tabelicy 3.7.

Tablica 3.7

*Czas tłumaczenia programów układanych w języku COBOL*

Honeywell			ICT		
Maszyna	Translator	Instrukcji/min.	Maszyna	Translator	Instrukcji/min.
H 120	B	100	1901 (8K)	Compact	42
H 200	D lub H	250	1901 (16K)	Compact	53
H 1200	D lub H	300	1903 (16K)	Compact	145
H 2200	D lub H	350	1905 (32K)	Compact	450

Dla IBM, jak dotąd, dane opublikowane są dla COBOL — wersji E, dla maszyny o zestawie pamięci operacyjnej o pojemności 32 K oraz dwóch jednostek dysków magnetycznych, modeli 2311. Przy takiej strukturze zestawu maszyny dla modeli: IBM 360/30 — liczba tłumaczonych instr./min. wynosi 294, IBM 360/40 — liczba tłumaczonych instr./min. wynosi 320.

### 3.7. Systemy operacyjne

System operacyjny składa się z kompletu programów, które:

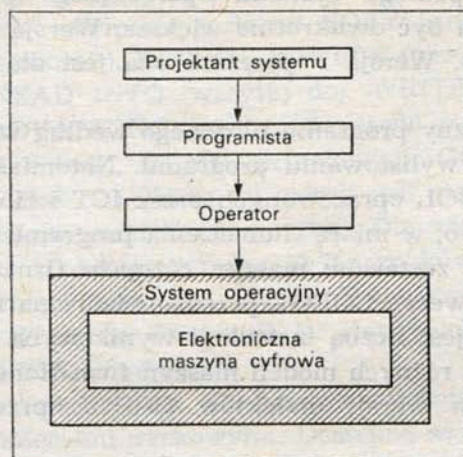
- pośredniczą w kontakcie operatora z maszyną,
- kierują wprowadzaniem danych i wyprowadzaniem wyników,
- kierują przetwarzaniem wewnątrz maszyny za pomocą translatorów, programów uniwersalnych (usługowych) i programów użytkowych.

System operacyjny jest zatem biblioteką specjalnych programów, które kierują maszynowym przetwarzaniem. Celem systemu operacyjnego jest

<sup>6</sup> Dane te pochodzą z pracy O. Shirley, *Comparing COBOL's, Data and Control Systems*, 1967, z. 1.



zwiększenie przepustowości przetwarzania na maszynie (*throughput*) oraz zmniejszenie czasu przetwarzania poszczególnych zadań obliczeniowych, a także ułatwienie obsługi maszyny. Na ryc. 3.4. przedstawiamy schemat



Ryc. 3.4. Związki między komputerem, systemem operacyjnym a użytkownikami

tycznie związki między systemem operacyjnym a maszyną i użytkownikami. System operacyjny umożliwia:

1) programiście:

- łatwy dostęp do często stosowanych programów i podprogramów,
- łatwe przechowywanie, wyszukiwanie, testowanie i modyfikowanie danych oraz programów,
- łatwe śledzenie bieżącego przebiegu przetwarzania danych,

2) operatorowi:

- stosowanie specjalnych rozkazów ułatwiających kierowanie przetwarzaniem,
- pośrednictwo w komunikowaniu się z maszyną,
- nadzór nad automatycznym wykonywaniem wszelkich funkcji ewidencyjnych związanych z eksploatacją systemów APD.

Pierwsze publikacje na temat systemu operacyjnego pochodzą z 1953 r., kiedy zaprezentowano je podczas Sympozjum na temat ETO zorganizowanego przez MIT<sup>7</sup>. Już wówczas, tak jak i obecnie, system operacyjny rozumiany był jako nieprzerwany ciąg operacji nad ciągiem wielu zadań (*job*), realizowany za pomocą przygotowanej dla maszyny biblioteki programów uniwersalnych. Pierwsze systemy operacyjne powstały dla maszyn IBM 709 i IBM 704 przy współpracy IBM i grupy SHARE, stąd nazwa tego systemu — SOS (*Share Operating System*). Jednym z pier-

<sup>7</sup> Por. C. W. Adams, I. H. Laning, Jr., *The MIT Systems of Automatic Coding: Comprehensive, Summer Session, Algebraic*, 1954, Office of Naval Research, Department of the Navy.



wszycy bardziej znanych systemów operacyjnych był FMS (*FORTRAN Monitor System*). Systemy te można zaliczyć do pierwszej generacji systemów operacyjnych. Charakteryzuje je zminimalizowanie czasów przygotawczo-zakończeniowych w przetwarzaniu partiowo-okresowym. Dla pierwszej generacji systemów operacyjnych typowe jest to, że ciąg zadań przetwarzania ma do swojej dyspozycji całą maszynę, a programy je realizujące przechowywane są w pamięci operacyjnej podczas całego okresu przetwarzania.

Dla pierwszej generacji typowa kolejność przetwarzania była następująca:

program źródłowy — tłumaczenie — program wynikowy — maszyna. Równocześnie rozwijał się drugi kierunek budowy systemów operacyjnych przystosowany do działania w czasie na bieżąco. Można wymienić pewne systemy APD, które przyczyniły się do rozwoju tego kierunku: SAGE, MERCURY, SABRE. Można dodać, że dla wojskowego systemu SAGE — system operacyjny pod nazwą Utility Control Program opracowało Laboratorium im. A. Lincolna.

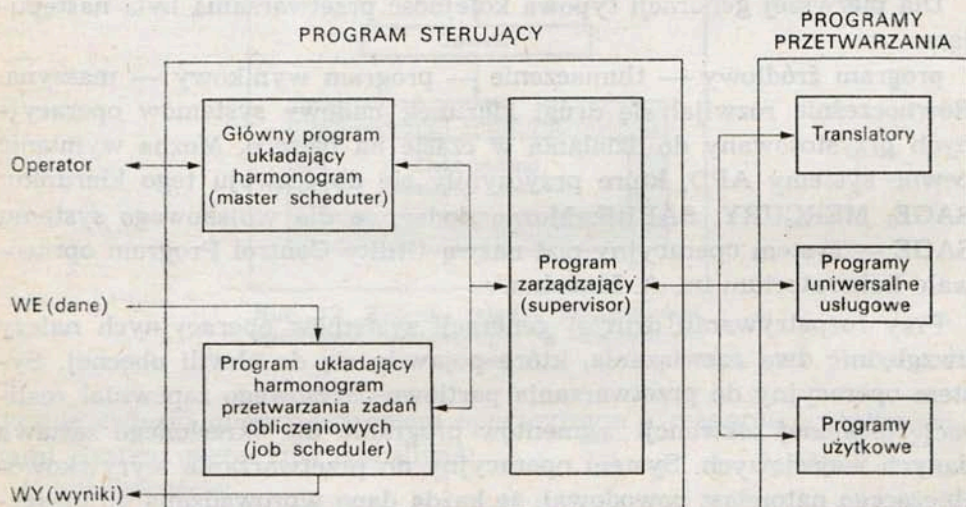
Przy rozpatrywaniu drugiej generacji systemów operacyjnych należy uwzględnić dwa rozwiązania, które pojawiły się do chwili obecnej. System operacyjny do przetwarzania partiowo-okresowego zapewniał realizację ustalonej sekwencji segmentów programu dla określonego zestawu danych wejściowych. System operacyjny do przetwarzania wyrwykowo-bieżącego natomiast powodował, że każda dana wprowadzana do maszyny w zależności od jej rodzaju wywoływała od razu na bieżąco odpowiedni program lub kilka programów. W obu wypadkach występowały pewne kombinacje programów i danych, których własności zostały wykorzystane dopiero w systemie następnej generacji. Kombinacje programów i danych nazwano w systemach operacyjnych trzeciej generacji zadaniami (*task*); zadania o podobnym charakterze grupuje się w celu łącznego przetwarzania w dogodnej kolejności, przez co osiąga się lepsze wykorzystanie urządzeń maszyny. W systemie operacyjnym maszyny IBM 360 istnieje wyspecjalizowana grupa programów do kierowania zadaniami (*management task*).

Przykładem dwóch zadań może być przetwarzanie dwóch różnych danych wejściowych za pomocą jednego programu. Pierwsza dana i program stanowią jedno zadanie; druga dana i program stanowią również jedno zadanie. Występuje tu jeden program i dwa zespoły danych. Można podać przykład dwóch różnych programów i jednego zespołu danych zorganizowanych w dwa zadania, kiedy z jednej kartoteki należy wydrukować dwa różne zestawienia. Jest to zatem nowy rodzaj przetwarzania, tzw. przetwarzanie wielozadaniowe (*multitask operation*). W takim ujęciu mamy do czynienia z wieloprogramowością, kiedy jedna jednostka centralna maszyny przetwarza kilka zadań. Natomiast wieloprzetwarza-



nie ma miejsce, kiedy poszczególne zadania realizowane są przez oddzielne jednostki centralne jednego zespołu maszyny.

Rozpatrzmy bliżej oprogramowanie maszyn serii IBM 360 (por. ryc. 3.5), które składa się z programów tłumaczących (tłumaczy), programów uniwersalnych (usługowych), programów użytkowych oraz programu sterującego. Maszyny IBM 360 są zaopatrzone w tłumacze dla języków COBOL, FORTRAN, PL/I, RPG (Report Program Generator Language) oraz dla języka montującego (assembly language).



Ryc. 3.5. Struktura oprogramowania maszyn typu IBM 360

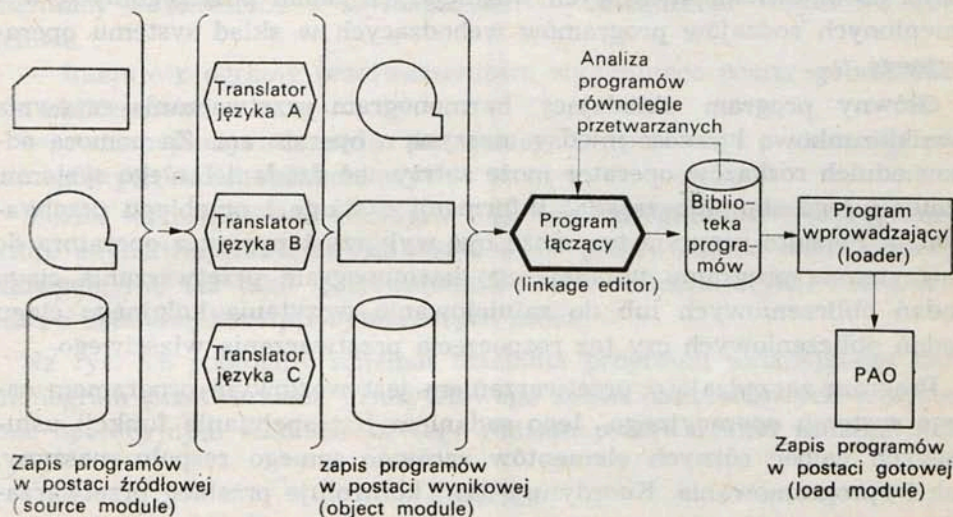
Do programów uniwersalnych usługowych zalicza się: programy modyfikujące zawartość biblioteki programów (przepisywanie, wydruki, korekta indeksu itp.), generator sortowania, dobierania oraz tzw. programu łączącego (*linkage editor*). Szczególne zadanie w systemie oprogramowania spełnia ten ostatni program. Właśnie jego funkcja jest jedną z dość charakterystycznych cech III generacji systemów operacyjnych.

Tłumaczenie programu źródłowego na program wynikowy może odbywać się niezależnie dla kilku programów (tak było w I generacji systemów operacyjnych), z tym jednak zastrzeżeniem, że program wynikowy w wypadku III generacji systemu operacyjnego nie ma przydzielonych rzeczywistych miejsc pamięci, wobec czego nie nadaje się jeszcze do wykonania przez maszynę. Doprowadzenie go do stanu nadającego się do wykonania (*load mode*) uwarunkowane jest strukturą pozostałych programów, które będą równolegle przetwarzane. Analizowania tej struktury oraz przydzielania rzeczywistych miejsc pamięci dokonuje program łączący tak, aby nie dopuścić do interferencji.

Na ryc. 3.6 przedstawiamy schemat działania programu łączącego. Zaczniemy od tego, że programy źródłowe mogą być umieszczone na

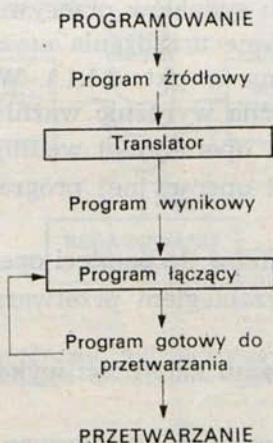


maszynowych nośnikach informacji w dowolnym czasie, również w dowolnym czasie mogą być przetłumaczone na programy wynikowe i umieszczone na maszynowych nośnikach informacji. Jednakże przydzielenie tym programom wynikowym określonych miejsc pamięci oraz wybranie



Ryc. 3.6. Schemat działania programu łączącego różne programy w maszynie typu IBM 360

programów wynikowych do współbieżnego przetwarzania odbywa się przy udziale programu łączącego, którego miejsce w całym cyklu programowania—przetwarzanie ilustruje ryc. 3.7.



Ryc. 3.7. Miejsce programu łączącego w cyklu programowania — przetwarzanie

Zgodnie ze schematem przedstawionym na ryc. 3.5 program sterujący, który zgodnie z przyjętą nomenklaturą jest właściwym systemem opera-



cyjnym maszyny IBM 360 składa się z: głównego programu układającego harmonogram przetwarzania (*master scheduler*), programu układającego harmonogram przetwarzania ciągu zadań obliczeniowych (*job scheduler*) oraz programu zarządzającego przetwarzaniem (*supervisor*). Przejdźmy teraz do omówienia niektórych ważniejszych zadań poszczególnych wymienionych rodzajów programów wchodzących w skład systemu operacyjnego.

Główny program układający harmonogram przetwarzania zapewnia dwukierunkową łączność między maszyną a operatorem. Za pomocą odpowiednich rozkazów operator może zatrzymać działanie całego systemu, zmienić jego stan lub zażądać informacji o stanie i przebiegu przetwarzania. Ponadto program ten może być wykorzystany przez operatora do zatrzymania programu układającego harmonogram przetwarzania ciągu zadań obliczeniowych lub do zainicjowania wczytania kolejnego ciągu zadań obliczeniowych czy też rozpoczęcia przetwarzania właściwego.

Program zarządzający przetwarzaniem jest węzłowym programem całego systemu operacyjnego. Jego zadaniem jest spełnianie funkcji usługowych wobec różnych elementów zarówno samego zespołu maszyny, jak i oprogramowania. Koordynuje on i kontroluje przebieg przetwarzania w ten sposób, aby możliwie najefektywniej wykorzystać dostępne rozwiązania konstrukcyjno-programowe maszyny. Między innymi zabezpiecza przed interferencją programów przetwarzaniowych i programów systemu operacyjnego. W tym celu korzysta z uprzywilejowanych rozkazów tylko dla systemu operacyjnego, takich, jak np. rozkazy zabezpieczające pamięć, rozkazy wejściowo-wyjściowe. Spełniając tego typu funkcje korzysta z systemu sygnałów przerywających przetwarzanie wysyłanych zarówno przez same urządzenia maszyny, jak i programy (system tego typu był opisany w pkt. 2.1.1.). Wśród wielu funkcji, które realizuje ten program, można wyróżnić ważniejsze, a mianowicie:

- przydzielanie pamięci operacyjnej według wymagań programów,
- przydzielanie pamięci operacyjnej programom w zależności od kolejności ich przetwarzania,
- wprowadzanie programów do pamięci operacyjnej,
- nadzorowanie nad przebiegiem przetwarzania według wprowadzonych programów,
- układanie harmonogramu kolejności wykonywania operacji wejściowo-wyjściowych,
- nadzorowanie, by programy wyznaczone na dany dzień i godzinę zostały wykonane,
- ewidencjonowanie zużycia czasu maszyny w różnych układach sprawozdawczych,
- ewidencjonowanie dla celów konserwacji — różnych wypadków błędów, uszkodzeń i awarii itp.

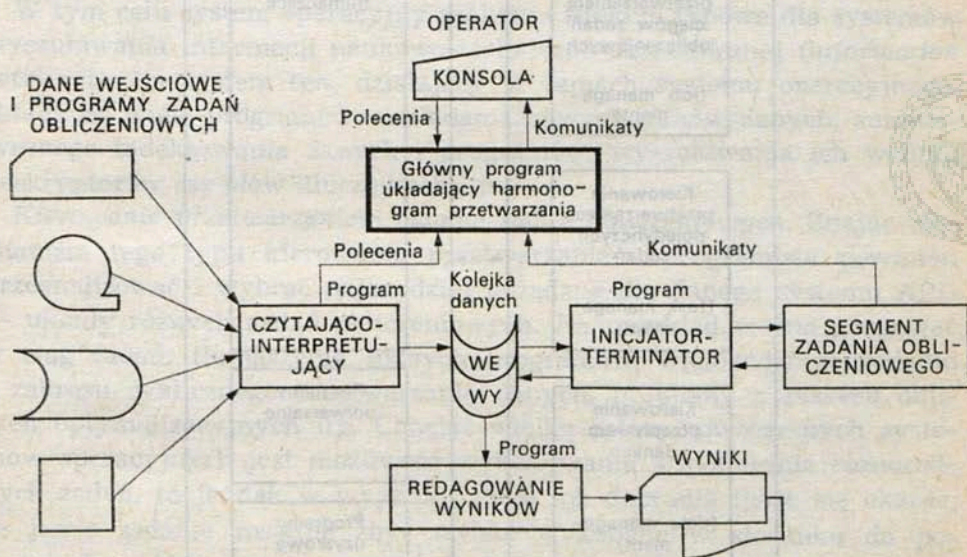


Program układający harmonogram przetwarzania ciągu zadań obliczeniowych realizuje następujące funkcje:

- analizuje wymagania poszczególnych zadań obliczeniowych,
- przydziela poszczególnym zadaniom obliczeniowym (lub ich segmentom) — określone w wymaganiach — urządzenia wejściowo-wyjściowe.
- inicjuje programy przetwarzaniowe wykonujące poszczególne zadania obliczeniowe.
- ewidencjonuje niezbędne informacje o przebiegu wykonania poszczególnych zadań obliczeniowych.

Można wyróżnić kilka rodzajów tego typu programów. Są programy, które umożliwiają kolejne lub współbieżne przetwarzanie zadań obliczeniowych, lub też takie przetwarzanie zadań obliczeniowych, które wynika z ustalonej listy priorytetów tych zadań.

Na ryc. 3.8 podajemy schemat działania programu układającego harmonogram przetwarzania priorytetowego zadań obliczeniowych w systemie operacyjnym IBM 360. W tego rodzaju przetwarzaniu zadania obli-



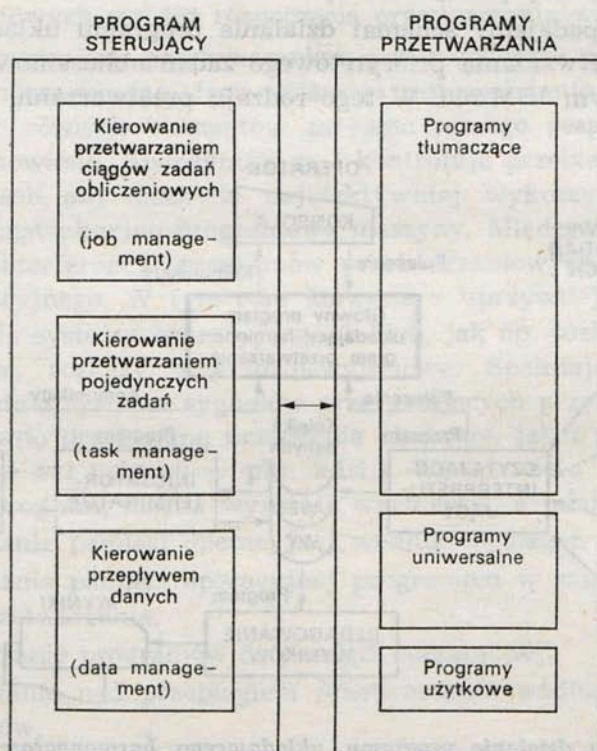
Ryc. 3.8. Schemat działania programu układającego harmonogram przetwarzania priorytetowego zadań obliczeniowych w systemie operacyjnym maszyn typu IBM 360

zeniowe nie są realizowane w tej samej kolejności, w jakiej są wprowadzane do maszyny. Z tego względu dane wejściowe i wyjściowe przechowywane są w pamięci zewnętrznej zwykle o wyrwykowym dostępie i czekają kolejno na przetworzenie. Program czytająco-interpretujący rejestruje zgłoszenia urządzeń wejściowych i wprowadza dane do pamięci masowej według ustalonej listy priorytetów. Z tej samej listy ko-



rzysta inny program, tzw. inicjator-terminator, który, przeglądając dane i wyniki oczekujące w kolejce, steruje wykonaniem poszczególnych zadań obliczeniowych zgodnie z ich ustaloną kolejnością. Zarówno program wprowadzająco-interpretujący, jak i inicjująco-terminujący przesyłają komunikaty do głównego programu układającego harmonogram przetwarzania.

Na zadania systemu operacyjnego zwróciliśmy dotychczas uwagę ze względu na funkcje realizowane przez poszczególne ważniejsze programy tego systemu. Natomiast dość istotnym zagadnieniem jest również syntetyczna ocena działania systemu operacyjnego z punktu widzenia wymagań projektanta systemu. Pod tym względem system operacyjny, na przykładzie programu sterującego maszyną IBM 360, realizuje trzy główne funkcje (por. ryc. 3.9):



Ryc. 3.9. Program sterujący maszyną typu IBM 360 i jego funkcje

a) kierowania przepływem danych (*data management*), które polega na tym, że wszystkie typy danych i programów przetwarzanych w maszynie są usystematyzowane, odpowiednio zorganizowane i identyfikowane, przechowywane, skatalogowane i odpowiednio modyfikowane, aktualizowane i wyszukiwane,



b) kierowania przetwarzaniem ciągów zadań obliczeniowych (*job management*), które polega na tym, że ciąg tych zadań przetwarzany jest w możliwie najbardziej efektywny sposób oraz przy minimalnym udziale operatora,

c) kierowania przetwarzaniem pojedynczych zadań (*task management*), które polega na tym, że odpowiednie zadania są odpowiednio grupowane w celu współbieżnego ich wykonania, ze względu na najlepsze wykorzystanie urządzeń maszyny.

*Kierowanie przepływem danych.* W nowoczesnych systemach przetwarzania danych jednym z ważniejszych problemów organizacyjnych jest sprawne kierowanie wielką masą danych i dużą liczbą programów. Pod tym względem system operacyjny umożliwia programiście:

- modyfikowanie danych i programów umieszczonych w masowych pamięciach zewnętrznych o wrywkowym dostępie,

- swobodne korzystanie i wymianę programów pisanych przez różnych programistów, a przechowywanych w maszynie,

- posługiwanie się usystematyzowanymi metodami identyfikowania i odnajdywania danych i programów będących w maszynie.

W tym celu system operacyjny realizuje funkcje typowe dla systemów wyszukiwania informacji naukowo-techniczno-ekonomicznej (*information retrieval*). Na system ten, działający w ramach systemu operacyjnego, składa się wiele programów: zakładania nowych plików danych, automatycznego indeksowania danych i programów, wyszukiwania ich według deskryptorów czy słów kluczowych itp.

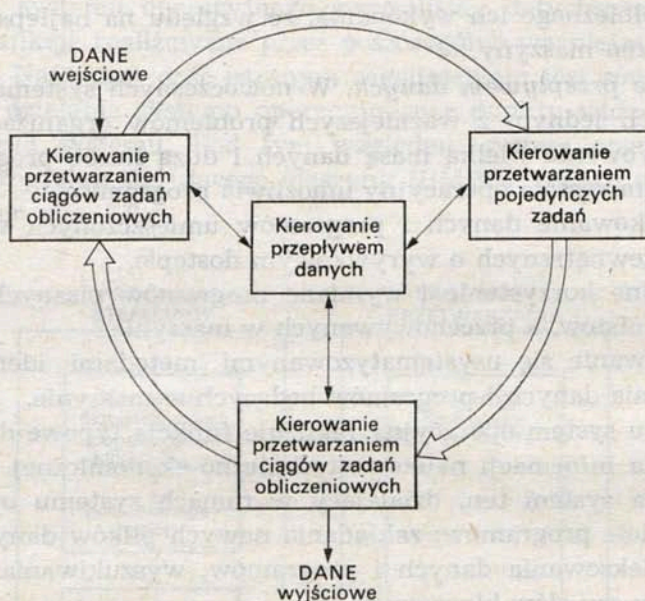
*Kierowanie przetwarzaniem ciągów zadań obliczeniowych.* Znając mechanizm tego typu kierowania przetwarzaniem, programista powinien przeanalizować i wybrać najbardziej pożądane dla danego systemu APD — układy różnych zadań obliczeniowych. Na przykład można grupować w ciąg zadań: tłumaczenia różnych programów, wielogodzinne zadania z zakresu cyklicznego przetwarzania danych, problemy z zakresu obliczeń optymalizacyjnych itp. Chociaż wielką zaletą nowoczesnych systemów operacyjnych jest możliwość wykonywania współbieżnie różnorodnych zadań, to jednak w wypadku złego ich dobrania może się okazać, że jedno zadanie zostanie zbyt szybko wykonane w stosunku do pozostałych, a tym samym spowoduje powstanie nowej rezerwy przetwarzaniowej dla nowych zadań, których nie przygotowano do przetwarzania.

*Kierowanie przetwarzaniem pojedynczych zadań.* Pod tym względem wpływ programisty na zaprojektowanie procesu przetwarzania jest znikomy. Omawiany system kierowania przetwarzaniem można porównać do problematyki partii ekonomicznej półfabrykatów występującej w procesie produkcyjnym. W obu wypadkach chodzi o zgrupowanie podobnych elementów (zadań), które można współbieżnie wykonać wykorzystując jednorazowo uniwersalny program (np. wejściowo-wyjściowy) wywołany



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,