2

Struktura automatycznego przetwarzania danych

2.1. Przetwarzanie wewnętrzne

Znajomość przez projektanta systemu, choćby tylko w ograniczonym zakresie, zagadnień organizacji przetwarzania wewnątrz komputera jest niezbędna. Przede wszystkim znajomość ta jest konieczna przy projektowaniu lub dobieraniu z biblioteki programów odpowiednich bardziej złożonych uniwersalnych programów maszyny oraz wyboru samego komputera maszyny matematycznej odpowiedniego do zrealizowania zadań wynikających z projektu systemu przetwarzania danych. Ponieważ komputer jest tylko środkiem technicznym występującym w systemie przetwarzania danych, zbyt dokładne wnikanie w zagadnienia konstrukcyjne jej budowy przez projektantów systemu nie jest celowe. Może to często prowadzić, co zresztą wynika z dotychczasowej praktyki, do wyolbrzymiania zagadnień technicznych maszyny w stosunku do zagadnień systemu przetwarzania danych. Warto podkreślić, że organizacja przetwarzania wewnątrz maszyny, poza pewnymi wybranymi, specjalnymi SPD, ma stosunkowo mniejszy wpływ na koncepcję budowy SPD od organizacji przetwarzania zewnętrznego. Z tych względów w zagadnieniach organizacji przetwarzania wewnątrz maszyny — zwrócimy uwagę tylko na węzłowe rozwiązania, które dotyczą:

— sposobu realizowania operacji maszynowych,

— możliwości przetwarzania kilku różnych programów na jednej maszynie w tym samym czasie,

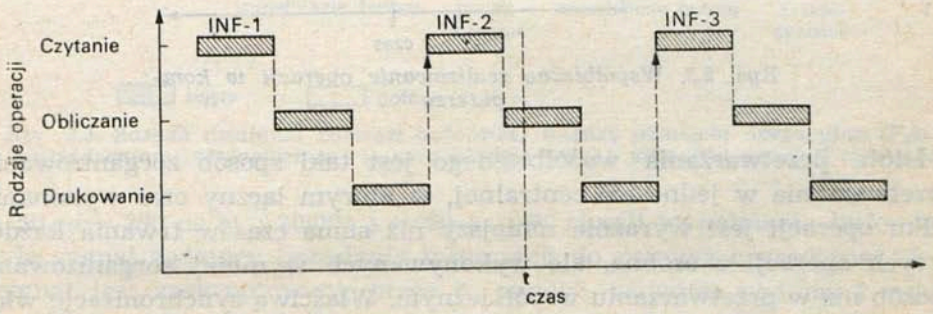
— możliwości alokacyjnych pamięci operacyjnej.

Rozwiązania te mają bezpośredni wpływ na budowę systemu operacyjnego maszyny i decydują w znacznym stopniu o skuteczności przetworzeniowej maszyny. We współczesnych komputerach rola systemu operacyjnego znacznie wzrasta. Zły system operacyjny może „spowolnić" maszynę zbudowaną z najszybszych układów elektronicznych i odwrotnie.

2.1.1. *Posobne a współbieżne realizowanie operacji*

Zagadnienie współbieżnego przetwarzania przedstawimy na przykładzie zespołu bardzo małej maszyny, złożonego oprócz jednostki centralnej tylko z czytnika kart i drukarki. Maszyna realizuje proste przetwarzanie, polegające na wczytaniu kolejnej karty, wykonaniu odpowiedniego obliczenia i niezwłocznym wydrukowaniu wyniku. Na schematach grupę takich informacji jednokartowych oznaczymy symbolem INF, gdzie n jest numerem kolejnej karty. Otóż w takim wypadku można wyodrębnić dwa zasadniczo różne sposoby przetwarzania:

a) przetwarzanie posobne (*succesive*)[1] — operacje czytania, obliczania i drukowania dotyczące bieżącej karty odbywają się w tym sensie szeregowo, że ich realizacja następuje dopiero po zakończeniu cyklu przetwarzania karty poprzedniej, co powoduje zużytkowanie czasu pracy jednostki centralnej tylko w 33%,



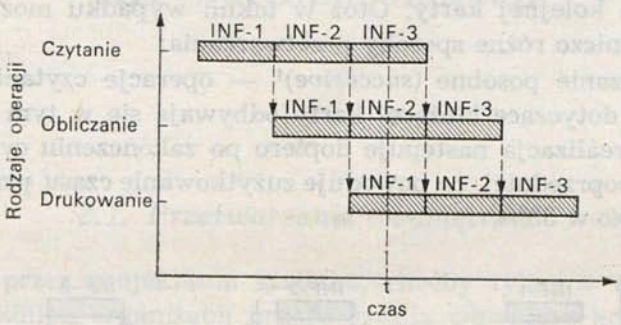
Ryc. 2.1. Posobne realizowanie operacji w komputerze

b) przetwarzanie współbieżne (*concurrently*) — operacje czytania, obliczania, drukowania dotyczące bieżącej karty odbywają się w tym sensie równolegle, że ich realizacja następuje jednocześnie z przetwarzaniem dwu poprzednich i dwu następnych kart, wskutek nakładania się (*overlapping*) cyklów przetwarzania, tak aby skoro tylko dane urządzenie zakończy operację odnoszącą się do karty n-tej, natychmiast przechodziło do wykonania operacji odnoszącej się do karty $n+1$ (por. ryc. 2.2). Przy przetwarzaniu współbieżnym jednostka centralna w danej chwili czasowej t realizuje wszystkie trzy funkcje naraz, podczas gdy przy przetwarzaniu posobnym w danej chwili można realizować tylko jedną funkcję. W rzeczywistości operacja przebiega nie zawsze tak idealnie, jak to przedstawiono na ryc. 2.2 i należy się liczyć z małymi stratami czasu.

---

[1] Często niesłusznie nazywane przetwarzaniem szeregowym. W przetwarzaniu szeregowym mamy do czynienia z określonym stałym taktem przetwarzania, który w danej sytuacji może być wypadkiem szczególnym, o ile czasy trwania operacji czytania, obliczania i drukowania dla kolejnych kart byłyby równe. Ponieważ najczęściej tak nie jest, operacje dla kolejnych kart odbywają się „po sobie" jakby asynchronicznie.

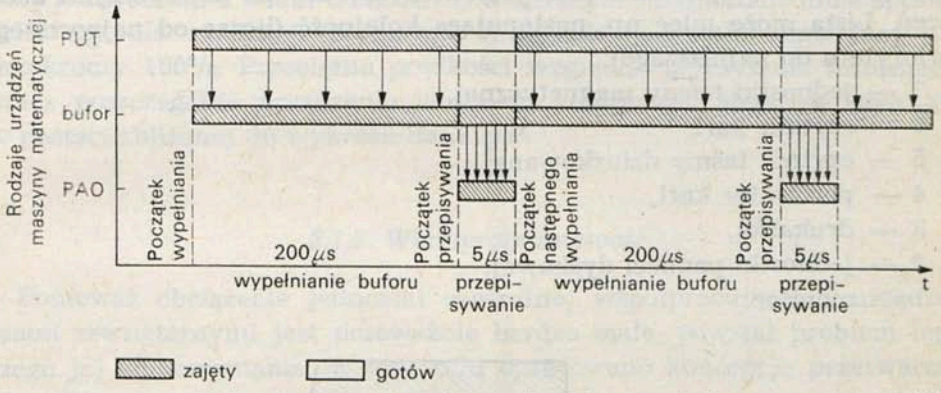
W nomenklaturze fachowej do lat 1963/1964 współbieżność przetwarzania była określana mianem „zdolności maszyny do podziału czasu" *(time sharing)*. Obecnie jednak, w związku z rozwojem koncepcji wielodostępności i wieloprogramowości, które wyjaśnimy, termin „podział czasu" nabrał szerszego znaczenia, przetwarzanie współbieżne odpowiada więc tylko „prostemu podziałowi czasu", gdyż „zaawansowany podział czasu" jest bardziej skomplikowany (ale i bardziej wydajny).



Ryc. 2.2. Współbieżne realizowanie operacji w komputerze

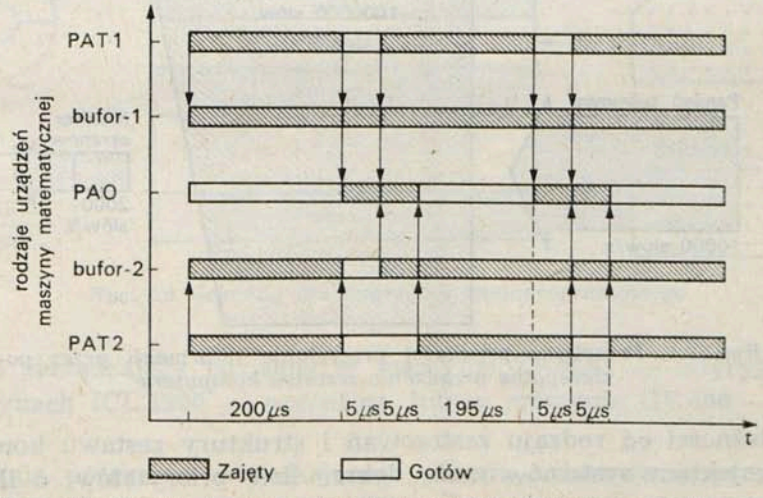
Istotą przetwarzania współbieżnego jest taki sposób zorganizowania przetwarzania w jednostce centralnej, w którym łączny czas wykonania kilku operacji jest wyraźnie mniejszy niż suma czasów trwania każdej z tych operacji z osobna, ale wykonywanych w mniej zorganizowany sposób niż w przetwarzaniu współbieżnym. Właściwą synchronizację włączania poszczególnych operacji w procesie obliczeniowym zapewnia specjalny program nadzorczy maszyny, zwany przerywnikowym albo przerywnikiem (*interrupt*). Wybór sekwencji operacji, które mają być wykonane w danym interwale czasowym, wynika jednoznacznie z wartości liczb przyjętych do określenia priorytetów kolejności. Liczby te przyporządkowane są poszczególnym częściom programów lub urządzeniom zewnętrznym; określają one pilność przetwarzania. Samo przerywanie polega więc na zawieszeniu realizacji bieżącej sekwencji rozkazów o danym priorytecie i rozpoczęciu wykonania innej sekwencji (lub powrocie do jeszcze innej sekwencji uprzednio zawieszonej) o wyższym priorytecie. Sygnałem do przerwania może być spełnienie określonego warunku logicznego w przetwarzanym programie albo zgłoszenie się któregoś urządzenia zewnętrznego o wyższym priorytecie. To ostatnie odbywa się na ogół np. w maszynie typu IBM 360 przez analizę słowa *(status word)* przyporządkowanego określonemu kanałowi zewnętrznemu. Każda cyfra (lub grupa cyfr) w słowie sterującym oznacza: operację zewnętrzną, rodzaj urządzenia zewnętrznego lub jego stan. Słowo takie jest badane, aby ustalić rodzaj przyszłej operacji i ustalić czy przerwanie powinno nastąpić już, czy też ulec odroczeniu.

Działanie przerywnika postaramy się wyjaśnić obrazowo na nieco bardziej skomplikowanym przykładzie w stosunku do poprzedniego. Jednostka pamięci taśmowej o prędkości przesyłania 20 000 znaków (*charakters*) na sekundę (w skrócie 20 *Kc*) jest przyłączona do pamięci operacyjnej o zdolności przesyłania 200 *Kc* przez tzw. pamięć buforową o pojemności 4 znaków; cykl pamięci operacyjnej wynosi 1 : 200 000 s = = 5 μs/znak. Wypełnienie takiego buforu nastąpi zatem w czasie



Ryc. 2.3. Zasada działania pamięci buforowej między pamięcią operacyjną (PAO) a urządzeniami zewnętrznymi na przykładzie jednej jednostki pamięci taśmowej

4·50 μs = 200 μs (1 : 20 000 s = 50 μs). W chwili wypełniania bufor wysyła sygnał żądający przesłania informacji do pamięci operacyjnej. Jeśli sygnał jest zaakceptowany przez tę pamięć, wówczas podczas 1 cyklu, tzn. w ciągu 5 μs nastąpi przesłanie. Obciążenie jednostki centralnej wyniesie wówczas około 2,5%. Ponieważ wypełnianie buforu trwa długo,

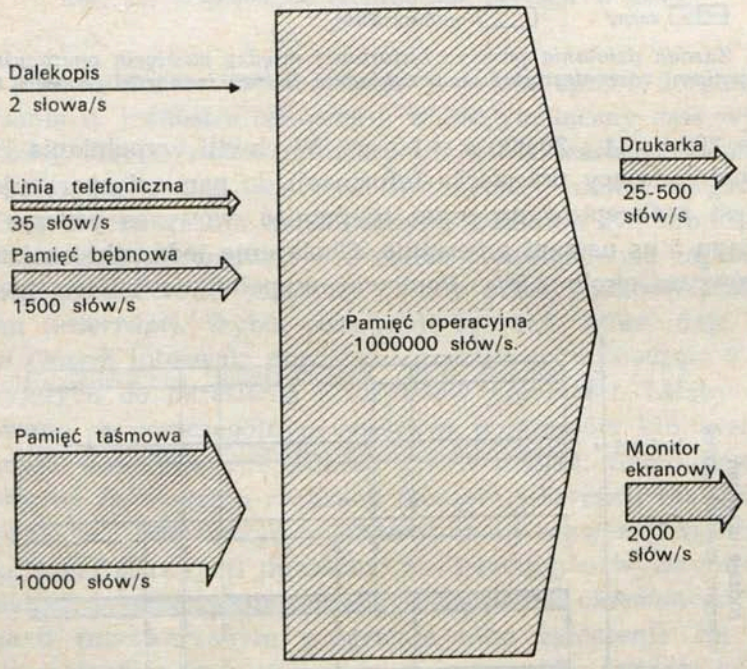


Ryc. 2.4. Zasada działania dwóch pamięci buforowych między pamięcią operacyjną (PAO) a dwoma jednostkami pamięci taśmowej (PAT)

jest pożądane, aby przesłanie z buforu do pamięci operacyjnej odbyło się możliwie najszybciej, gdyż bufor musi być przygotowany do następnego wypełnienia. Zatem w czasie napełniania buforu pamięć operacyjna jest wolna przez 195 μs i w tym czasie może realizować funkcje w połączeniu z innym urządzeniem zewnętrznym.

Jeśli naraz kilka buforów urządzeń zewnętrznych zgłosi gotowość komunikacji z pamięcią operacyjną, należy dokonać wyboru. Dla każdej maszyny ułożona jest lista priorytetów, począwszy od najszybszych urządzeń. Lista może mieć np. następującą kolejność (licząc od najwyższego priorytetu do najniższego):

7 — jednostki taśmy magnetycznej,
6 — czytnik kart,
5 — czytnik taśmy dziurkowanej,
4 — perforator kart,
3 — drukarka,
2 — jednostki pamięci dyskowej,
1 — monitor.



Ryc. 2.5. Przeciętne prędkości przesyłania informacji przez poszczególne urządzenia zestawu komputera

W zależności od rodzaju zastosowań i struktury zestawu komputerowego projektant systemów może dobrać listę priorytetów, o ile na to zezwalają rozwiązania organizacyjne danego systemu.

W zależności od typu maszyny i rodzaju urządzeń zewnętrznych ob-

ciążenie czasowe jednostki centralnej jest różne; np. dla modelu GE-435 wynosi:

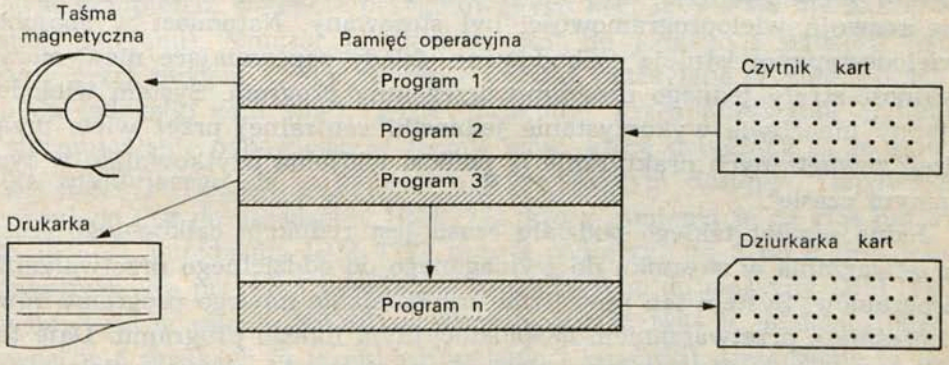
czytnik (900 kart/min.)	0,9%
dziurkarka (100 kart/min.)	1,2%
drukarka (1200 wierszy/min.)	1,5%
taśmy magnetyczne (41 *Kc*)	2,1%

W niektórych skomplikowanych systemach zastosowań, szczególnie przy korzystaniu z wielu urządzeń zewnętrznych, projektant musi sprawdzić, czy dla realizacji programu obciążenie jednostki centralnej nie przekroczy 100%. Przeciętne prędkości względne przesyłania informacji przez poszczególne urządzenia zestawu komputera ilustruje ryc. 2.5 w postaci zbliżonej do wykresu Sankeya[2].

2.1.2. Wieloprogramowość

Ponieważ obciążenie jednostki centralnej współpracującej z urządzeniami zewnętrznymi jest przeważnie bardzo małe, powstał problem lepszego jej wykorzystania. W tym celu opracowano koncepcję przetwarzania wieloprogramowego.

Wieloprogramowość oznacza równocześnie wykonywanie na tej samej maszynie kilku programów naraz; każdemu programowi przydziela się przy tym odrębne strefy pamięci operacyjnej i odrębne urządzenia zewnętrzne (por. 2.6). Wykonywanie tych programów koordynuje specjalny



Ryc. 2.6. Schemat przetwarzania wieloprogramowego

program zarządzający nazwany w maszynie ZAM 41 — „dyrygentem", w maszynach ICL 1900 — *executive*, lub w maszynie GE 400 — *supervisor*.

Działanie programu zarządzającego można zilustrować przykładem,

[2] Rycina pochodzi z pracy A. G. Fauret, *Introduction to Digital Computer Application*, Reinhold 1965, s. 64.

polegającym na równoczesnym wykonywaniu dwu programów P1 i P2. Jeżeli — przyjmijmy — w pewnym momencie przy przetwarzaniu programu P1 nastąpiło zarejestrowanie „gotowości do pracy", np. drukarki przydzielonej programowi P2, to wówczas program zarządzający:

a) pozwala zakończyć aktualnie wykonywany rozkaz o adresie n z programu P1,

b) umieści adres m czekającego na wykonanie rozkazu z programu P2 na pierwszym miejscu „kolejki",

c) umieści adres $n+1$ na drugim miejscu „kolejki",

d) przystępuje do realizacji „kolejki" wybierając rozkaz spod adresu m, a po wykonaniu tego adresu

e) dopuszcza zatrzymany rozkaz z „kolejki" (adres $n+1$) i wykonuje w dalszym ciągu program P1.

W systemie wieloprogramowym występuje również lista priorytetów realizacji poszczególnych programów. Wieloprogramowość może być realizowana sposobem:

— o ustalonym dostępie,

— wielodostępnym (*multi-acces*).

Te dwa sposoby wieloprogramowości różnią się tzw. systemem protekcji pamięci. Przetwarzanie naraz paru programów wymaga podziału pamięci operacyjnej na strefy, tak aby w czasie przetwarzania programy ani ich dane nie zachodziły na siebie. W sposobie o ustalonym dostępie protekcja pamięci realizowana jest przez program i nie mamy całkowitej pewności niezniszczenia poszczególnych stref pamięci operacyjnej. Ten sposób został już wprawdzie zarzucony, jednakże w początkowym okresie rozwoju wieloprogramowości był stosowany. Natomiast w sposobie wielodostępnym istnieją rozbudowane układy zapewniające niewymazywalność strefy jednego programu przez inny program. System wielodostępny umożliwia wykorzystanie jednostki centralnej przez wiele urządzeń zewnętrznych praktycznie (z punktu widzenia użytkownika) w tym samym czasie[3].

Jedną z zalet takiego podziału czasu jest redukcja całkowitego czasu przetwarzania w stosunku do wymaganego do oddzielnego przetwarzania programów. System ten umożliwia uruchamianie nowego programu równocześnie z przetwarzaniem eksploatacyjnym innego programu. Daje też całkowitą pewność, że programista nie znający aktualnego podziału pamięci operacyjnej na strefy, podczas uruchamiania „swojego" programu nie zniszczy zawartości „cudzych" miejsc pamięci. Podłączenie do maszyny kilku monitorów umożliwia równocześnie uruchamianie kilku nowych programów. Monitor podłączony do maszyny za pomocą linii trans-

[3] W rzeczywistości, jeśli maszyna nie ma co najmniej dwóch arytmometrów, to wówczas operacje poszczególnych programów wykonywane są w odpowiedniej kolejności. Niektóre maszyny, np. typu IBM 360, mają kanały wejściowo-wyjściowe typu selektora, w których pewne operacje dotyczące wprowadzania i wyprowadzania mogą zachodzić równocześnie z realizowaniem operacji w jednym arytmometrze

misji danych umożliwia nawet uruchamianie nowych programów na odległość.

System wielodostępny poważnie usprawnia organizację pracy w ośrodku obliczeniowym. Monitory można ustawić w pomieszczeniach programistów tak, że programista nie wychodząc z pokoju może uruchamiać programy. System wielodostępny może również być wykorzystywany przez personel kierowniczy.

Od dłuższego czasu prowadzone są badania nad uproszczeniem kontaktu człowieka z maszyną. Zbyt dużo jest jeszcze konwencji w obsłudze komputerów, aby uważać ich użytkowanie za proste. Najwygodniejszy byłby niewątpliwie kontakt głosowy (pewne prace prowadzi w tym zakresie np. firma BELL). Na razie można przedstawić etap pośredni, w którym odpowiedzi na pytanie maszyna daje w postaci głosowej, wybierając z taśmy magnetofonowej uprzednio nagrane przez spikera brzmienie poszczególnych liczb. System ten został opracowany przez Centrum Studiów i Badań francuskiej filii koncernu IBM w La Gaude.

Urządzenie o nazwie Audio Reponse-IBM 7772 umożliwia wyprowadzanie informacji z maszyny w postaci głosu ludzkiego. Wyprowadzanie następuje w odpowiedzi na pytanie postawione w postaci ciągu odpowiednio dobranych kodów bądź przez nakręcenie tarczy telefonu, wreszcie przez wprowadzanie kartą dziurkowaną z czytnika IBM 1001. Karta jest zwykle już wstępnie wydziurkowana ze stale powtarzającymi się kodami. Z pulpitu czytnika można dzięki klawiaturze wprowadzić dodatkowe kody. Dla najczęściej powtarzających się pytań można dysponować kompletem wstępnie przygotowanych kart (aby stale nie nakręcać tarczy telefonu). Monitor przyłączony jest do jednego z kanałów maszyny przez modem. Cyfry tworzące pytanie przesyłane są znak po znaku do jednostki centralnej, gdzie następuje przeanalizowanie pytania i sformułowanie odpowiedzi w formie słów, które dobierane są ze słownika znajdującego się w pamięci o wyrywkowym dostępie. Odpowiedź przesyłana jest do urządzenia IBM 7772, które zamienia je na głos ludzki. Do jednego urządzenia IBM 7772 można podłączyć od 2 do 8 linii telefonicznych. Na przykład urządzenie to podłączone do maszyny IBM 1440, wystawione w salonie SICOB w Paryżu, już w 1964 r. udzielało odpowiedzi w 5 językach (4 języki europejskie i japoński). Urządzenie to jest dalszym uproszczeniem monitora — maszyny do pisania. Dyrektor zamiast telefonować np. do działu zaopatrzenia w sprawie zapasu określonego asortymentu, do działu planowania w sprawie wskaźnika techniczno-ekonomicznego, czy też do działu zatrudnienia i płac w sprawie wydatkowanego funduszu płac — bierze słuchawkę telefonu innego aparatu i stosując odpowiedni system kodowy telefonuje po prostu... do komputera. Sytuacja taka może równie dobrze dotyczyć średniego personelu kierowniczego, jak i szczebli wyższych.

2.1.3. Alokacja pamięci operacyjnej

W systemie wieloprogramowym wypełnianie pamięci operacyjnej rozkazami, informacjami stałymi oraz wynikami pośrednimi poszczególnych programów jest bardzo skomplikowane. Już w systemie jednoprogramowym, przy niezależnym układzie kilku podprogramów, które mają później być powiązaną całością, trudno od razu ustalić właściwy przydział adresów. Trudności te jeszcze bardziej rosną przy układaniu licznych podprogramów, z których część lub nawet wszystkie naraz są przetwarzane współbieżnie. Stąd konieczność automatycznego przydzielania stref adresowych przygotowanym do przetwarzania programom. Żaden bowiem programista nie jest w stanie przewidzieć wszystkich możliwych kombinacji, w których występować będą jego podprogramy czy też programy w toku współbieżnego przetwarzania. Jest to tym bardziej oczywiste, że tłumaczenie poszczególnych programów napisanych w języku źródłowym na język maszyny wcale nie musi być wykonywane współbieżnie; w każdym natomiast wypadku należy zapewnić jednoznaczność etykiet, czyli tzw. adresów symbolicznych, które nie mogą być przyporządkowane tym samym adresom, o ile odpowiednie programy będą wykonywane współbieżnie.

Jeżeli pominąć pseudoalokację, jaka wynika z adresowania bezwzględnego jeszcze w toku układania programu, to na ogół należy wymienić tutaj dwie zasadnicze techniki automatycznego przydzielania programom odrębnych stref pamięci operacyjnej: alokację statyczną i dynamiczną.

Statyczna alokacja polega więc zatem na przeniesieniu roli koordynatora dysponującego pamięcią operacyjną z programisty na specjalny program łączący (*linkage-editor*). Program łączący przydziela programowi wynikowemu najbliższe ze stojących do dyspozycji strefy rezerwowe. Jedyna ingerencja programisty lub operatora polega na sprawdzeniu, czy dana kombinacja programów nie wymaga większej pojemności pamięci operacyjnej od tej, która jest dostępna. Określenie niezbędnej pojemności pamięci operacyjnej do realizacji odpowiedniej kombinacji programów nie zawsze jest możliwe. W wypadku organizacji wielodostępowej, kiedy z jednej maszyny korzysta naraz wielu użytkowników, jest to praktycznie niemożliwe. Jeśli wykonywanie określonego programu zostaje zawieszone, to program zostaje przesłany do pamięci rezerwowej (*dump*), z której ponownie zostaje przywołany po wykonaniu priorytetowych operacji. Program wraca, ale na swoim miejscu może napotkać program nowego użytkownika, zostaje zatem przesłany do innej części pamięci operacyjnej (obecnie wolnej), a nie do tej, którą przedtem zajmował.

W pewnych wypadkach można przy zachowaniu wielu ograniczeń dokonywać alokacji pamięci operacyjnej techniką statyczną (w wypadku priorytetowego przetwarzania). Aby to zrealizować, wymagana jest odpo-

wiednia segmentacja programu. Program dzieli się na segmenty danych i procedury, przy czym wymaga się, aby procedury nie modyfikować przez program lub zawartość komórek ulegających zmianie. Kiedy pojawi się sygnał przerwania, część programu z danymi zostaje skierowana do pamięci rezerwowej. Segment ten zawiera m.in. dane o stanie rejestrów maszyny w toku przetwarzania danego programu. Zawartość rejestrów w momencie przerwania nie może jednak zawierać adresów bezwzględnych. Czas przesyłania do pamięci rezerwowej może być zredukowany, jeśli segment danych można podzielić na podsegmenty danych odczytywalnych i zapisywalnych, wówczas bowiem tylko pierwszy podsegment byłoby celowe przesłać do pamięci rezerwowej. Przy spełnieniu tych warunków program łączący mógłby przeadresowywać program powracający do realizacji po przerwaniu. Przeadresowanie dotyczyłoby oryginalnej postaci segmentu procedury oraz ostatniej postaci segmentu danych. Pisanie programów podzielonych na segmenty danych i segmenty procedury nie jest łatwe, jednak w nowoczesnym programowaniu metoda ta zaczyna przeważać. Między innymi z tych względów odpada konieczność pośredniego adresowania, która zakłada przechowywanie adresów bezwzględnych w formie słownika.

Inny sposób może polegać na przyjęciu konwencji, że program może być przerywany tylko w określonych punktach. Wszystkie przerwania, które pojawią się przed tymi punktami, zostają zapamiętane i realizowane dopiero po osiągnięciu danego punktu programu.

Podstawową wadą statycznej alokacji pamięci operacyjnej jest złożony schemat harmonogramu przetwarzania, którego każdorazowe zoptymalizowanie zajmuje programowi łączącemu wiele czasu.

Dynamiczna alokacja. Przydział adresów programowi odbywa się dopiero w momencie jego realizacji, przy czym ta część pamięci operacyjnej zostaje przydzielona, która w danym momencie jest wolna, bez uwzględniania poprzedniej alokacji programu. Taka metoda nie wymaga segmentowania programu. Dynamiczna alokacja jest pożądana w sytuacjach, kiedy spora liczba programów jest równocześnie przechowywana w pamięci operacyjnej. Wiąże się to również z ochroną tych pól pamięci operacyjnej, w których występują niezależne programy. Dynamiczna alokacja może być realizowana przez program lub układowo. W pierwszym wypadku powstają specjalne wymagania co do postaci rozkazu, która musi przewidywać nie przetłumaczone adresy.

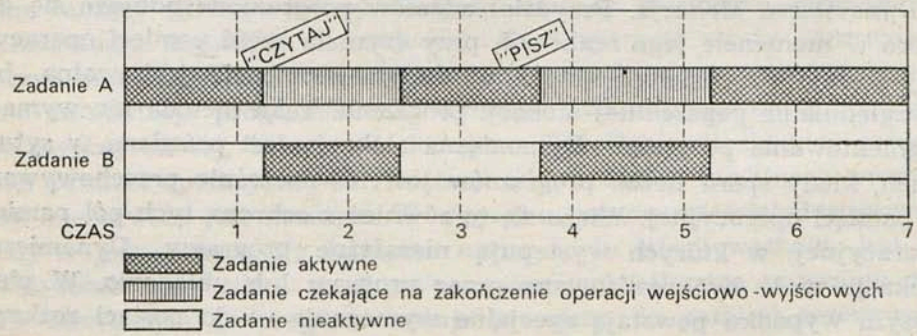
Maszyna IBM 360/67 posiada w adresie rozkazu 24 bity, co umożliwia określenie 16 milionów różnych adresów bajtowych, to jest wielokrotnie więcej, niż wynosi maksymalna pojemność pamięci operacyjnej. Ogół tych 16 mln adresów emulacyjnych podzielony jest na 16 segmentów, z których każdy dzieli się na 256 płatów o pojemności 4096 bajtów[4]

[4] Bajt składa się z 8 bitów i nazywany jest również oktetem.

każdy. Program łączący przydzielą danemu programowi tylko te płaty, które istnieją fizycznie w danej konfiguracji maszyny. Dwadzieścia cztery bity kodu adresowego podzielone są na trzy grupy: 4 bity określają segment, 8 bitów określa numer płatu, zaś 12 bitów określa numer bajtów w płacie. Tylko dwie pierwsze grupy bitów uczestniczą w alokacji dynamicznej. W ośmiu specjalnych rejestrach ewidencjonuje się dostępne segmenty i płaty pamięci. Z chwilą wykrycia dostępnych miejsc następuje w ciągu 200 μs przekształcenie na adres rzeczywisty segmentu i płatu, przy czym adres bajtowy pozostaje bez zmian.

2.1.4. Podział czasu

Rozwój konstrukcji dużych komputerów do obliczeń numerycznych spowodował z kolei rozwój systemów obliczeń dostępnych dla wielu (od kilkudziesięciu do kilkuset) użytkowników liczących współbieżnie na tym samym komputerze, ale z odległych miejsc, w których zainstalowane są teledatory albo tzw. końcówki (*terminal*) w postaci dalekopisów czy monitorów ekranowych (*display*). Stąd systemy te nazwano wielodostępnymi albo aboneckimi. Każdy użytkownik dysponuje określonym odcinkiem czasowym procesora zwanym kromką czasu (*time slice*). Licząc na komputerze użytkownikowi wydaje się, że tylko on w danej chwili korzysta z tego komputera, podczas gdy w rzeczywistości współbieżnie liczy nawet kilkaset innych użytkowników. Przydzielanie czasu użytkownikowi może odbywać się według zasady przyjętej w przetwarzaniu wieloprogramowym albo według zasady „kromkowania czasu".

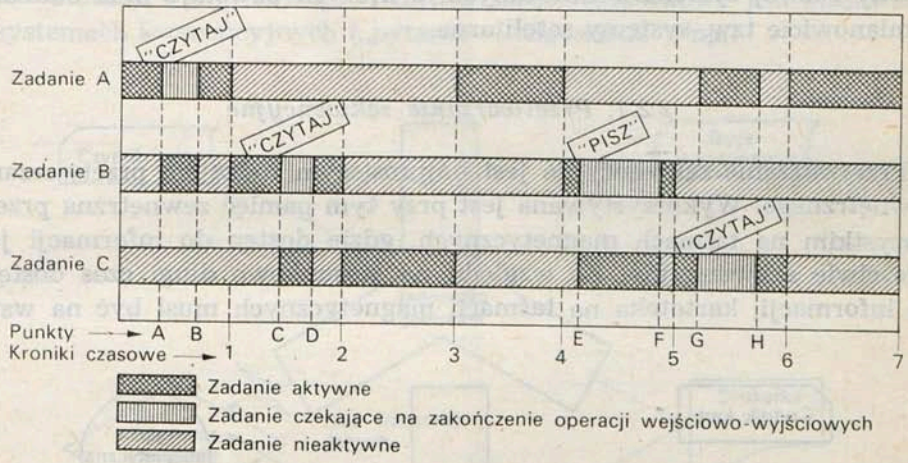


Ryc. 2.7. Podział czasu według zasady wieloprogramowości

Zasada podziału czasu przyjęta w przetwarzaniu wieloprogramowym (por. ryc. 2.7) polega na nieregularnym udostępnianiu procesora poszczególnym użytkownikom. Z przykładu podanego na ryc. 2.7 wynika, że zadanie A ma wyższy priorytet i w momencie czytania danych np. z czytnika czy pamięci zewnętrznej komputera zadanie B jest dopuszczone do obliczeń w procesorze. Jednak z chwilą zakończenia operacji

czytania w zadaniu A następuje przerwanie liczenia zadania B i ponownie zadanie A otrzymuje dostęp do procesora.

Zasada kromkowania czasu (por. ryc. 2.8) polega na przydzieleniu każdemu zadaniu jednakowego odcinka czasu. Zmiana użytkownika procesora może nastąpić w ramach kromki czasu tylko podczas realizowania



Ryc. 2.8. Podział czasu według zasady kromkowania

operacji na urządzeniach zewnętrznych komputera tak, jak ma to miejsce w pierwszej kromce czasowej w odniesieniu do zadania A i B. Już w drugiej kromce czasowej do procesora zostaje dopuszczone zadanie B, które również w toku czytania oddaje na chwilę dostęp do procesora zadaniu C.

2.2. *Przetwarzanie zewnętrzne*

Dobór warunków przetwarzania zewnętrznego odpowiednio do organizacji systemu APD jest podstawowym zagadnieniem w technologii maszynowego przetwarzania danych. Termin „przetwarzanie zewnętrzne" wynika ze stosowania nazwy „urządzenia zewnętrzne". Można wyróżnić dwa podstawowe rodzaje przetwarzania zewnętrznego, które dotyczą sposobu dostępu do większych zbiorów danych, np. do kartotek, przetwarzanie sekwencyjne i wyrywkowe. Oba rodzaje przetwarzania zewnętrznego APD mogą być realizowane sposobem: pośrednioautomatycznym lub bezpośrednioautomatycznym.

Przy przetwarzaniu sekwencyjnym wprowadzone dokumenty przed przetworzeniem ulegają odpowiedniemu posortowaniu, stosownie do formy prowadzonych kartotek przechowywanych w pamięci zewnętrznej. Przy przetwarzaniu wyrywkowym dokumenty są przetwarzane w kolejności ich napływu, bez uprzedniego sortowania, co umożliwia wyrywkowy dostęp do kartotek.